UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

ONLINE CONTEXTUAL UPDATING IN MULTI-CAMERA SCENARIOS

Alejandro López Cifuentes
Director: Marcos Escudero Viñolo
Supervisor: Jesús Bescós Cano

-MASTER THESIS-

Electronics Technology and Communications Department
Escuela Politécnica Superior
Universidad Autónoma de Madrid
June 2017

ONLINE CONTEXTUAL UPDATING IN MULTI-CAMERA SCENARIOS

Alejandro López Cifuentes

Director: Marcos Escudero Viñolo
Supervisor: Jesús Bescós Cano

Video Processing and Understanding Lab
Informatics Engineering Department
Escuela Politécnica Superior
Universidad Autónoma de Madrid
June 2017

This work has been partially suported by Ministerio de Economía, Industria y Competitividad of the Spanish Government and Fondo Europeo para el Desarrollo Regional of the European Union under the project TEC2014-53176-R (HAVideo)

Resumen

Este proyecto describe un sistema para realizar detección de personas y segmentación semántica en un escenario multicámara. La fusión de estas dos disciplinas tendrá como resultado detecciones de personas contextualmente filtradas. Se ha usado un entorno multicámara para reproyectar detecciones de una cámara a otra. Como ejemplo de uso, datos estadísticos de áreas semánticas concretas en la escena han sido igualmente extraídos. Para lograr una interacción completa con el usuario se ha diseñado una Interfaz de Usuario Gráfica a partir de una aplicación multihilo qué permitirá al usuario definir el entorno de detección de personas, así como mostrar resultados en tiempo real de ejecución.

Para poder llevar a cabo estos objetivos un estudio del estado del arte ha sido realizado. Se han analizado los diferentes detectores de personas, haciendo énfasis en aquellos que utilizan propuestas de objetos. Además, se han estudiado nuevos métodos en el entorno de la detección de personas tales como las redes neuronales. Se ha revisado el estado del arte actual sobre la extracción de información contextual, y en concreto, en el uso de la segmentación semántica. Finalmente, las ventajas y desventajas de los escenarios con configuración multicámara se han descrito.

Para lograr los objetivos mencionados, se ha propuesto un nuevo sistema que realiza detección de personas bajo diferentes condiciones de filtrado. Se han integrado en el sistema detectores como HOG, DPM, ACF, Fast-RCNN o PSP-Net y además se ha realizado segmentación semántica. Ambas fuentes de información han sido combinadas en un entorno común representado mediante un plano cenital de la escena.

Finalmente, el rendimiento del sistema ha sido probado en un data-set generado y manualmente anotado para generar graficas de rendimiento y estadísticos de uso.

Palabras Clave

Detección de personas, multi cámaras, segmentación semántica, plano cenital.

Abstract

This project describes a system to perform pedestrian detection and semantic segmentation in a multi-camera recorded scenario. The fusion of both disciplines leads to contextually filtered pedestrian detection. The multi camera system is used to reproject detections from one camera to the others. As an use example, statistical data usage of specific semantic areas in the scene is also extracted. For user interaction a Graphical User Interface (GUI) based on a multithread application has been designed. The GUI allows the user to define the method setup as well as display results in execution time.

In order to carry these tasks out a study of the state of the art has been done. Pedestrian detection approaches are reviewed emphasizing in those that rely on object proposals. Also, recent trends in the task of Pedestrian Detection are analyzed. In addition, current state of the art in the extraction of contextual information and −specifically− on the use of, semantic segmentation is studied. Finally, multi-camera scenarios are also described.

A new system has been proposed in order to achieve the objectives and perform pedestrian detections under different filtering and fusion conditions. Detectors such as HOG, DPM, ACF, Fast-RCNN or PSP-Net have been integrated and a complete semantic segmentation has been performed. Both information has been combined in a common developed frame.

Finally, system performance has been tested in a generated dataset with manually annotated ground truth.

Keywords

Pedestrian detection, multi camera, semantic segmentation, cenital plane.

Acknowledgements

En primer lugar, dar las gracias a mi tutor Marcos. Como durante el TFG, su continua ayuda a lo largo de este proyecto ha sido incalculable. Gracias por dejarme trabajar contigo de nuevo en el proyecto que más quería y por estar siempre disponible para cualquier problema. Ha sido todo un placer.

Gracias a todos esos compañeros de carrera que ahora más que nunca están presentes en el día a día y que han hecho que todos estos meses de duro trabajo sean más llevaderos.

Sin olvidar tampoco a todo el grupo del IPCV. Gente extrajera desconocida qué a base de convivencia y vivencias se han convertido en lo que es ya una gran familia que no olvidaré nunca.

Por último, mencionar a Elena, la persona que me ha animado continuamente y que ha hecho geniales estos dos últimos años.

Gracias a todos.

Alejandro López Cifuentes.

2017.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Nowadays, we live surrounded by electronic devices which claimed objective is to ensure the safety and security of the global population and to ease our lives on everyday tasks. These range from biometric systems [1] to all kind of different electrical sensors, including video surveillance cameras. These cameras are of real interest when developing Computer Vision algorithms in the scope of video surveillance [2].

The combination of these veins of research may lead to the automation of high-level human semantic tasks such as people detection [3], object detection and recognition [4, 5, 6] and extraction of contextual information [7]. The automation of these processes permits end-users build on these information sources to define the latest stages of video surveillance systems. These are usually the critical ones, e.g. alarm raising when some predefined event occurs.

Usually, video surveillance systems are focused either on the analysis of a single-camera point of view −which leads to a simple scenario in which the potential actions/events to detect are observed from a single point in the scene− or, on the analysis of a multi-camera setup. This last configuration may provide multiple benefits when analyzing big spaces as it provides to user different views of the scene, disambiguating occluded areas in the mono-camera views.

Among Computer Vision applications running on a multi-camera scenario, a pivotal field of research is the analysis of public spaces. These are often crowd populated scenarios which analysis requires the combination of the data obtained by all recording cameras. It is of real interest to analyze people behavior patterns [8, 9, 10] and temporal usage of a given area in large-scale scenarios such as shopping malls, universities and, generally, public-use buildings. Analysis ranges from the extraction

of statistical measures of behavior to the detection of anomalous unexpected events [11]. This results may come from a combination of complementary algorithms such as contextual and semantic area classification, people detection and crowd behavior analysis.

1.2 Objectives

The main objective of this thesis is to extract contextual descriptions from a large-scale populated multi camera scenario. A potential application of this task is illustrated by the extraction of temporal statistical usage data from relevant areas in the scene. The whole solution needs to be controlled through the use of a Graphical User Interface application.

To fulfill this objective, this work embraces two different blocks of objectives that complement each other. The first one targets the design of a graphical user interface (GUI). The second block deals with algorithm and research-related objectives.

Graphical User Interface

The GUI should be able to visualize and dynamically arrange −under a user-friendly environment− statistics from different areas of interest in a public space.

Algorithm

The algorithm related objectives are:

1. To integrate a semantic segmentation algorithm to perform contextual element in video sequences. The objective is to detect, classify and determine the spatial extend on each frame of the video of relevant elements such as doors, chairs, corridors and floor areas. We aim to:
   (a) Combine semantic information coming from different cameras.
   (b) Identify the usage rate of some important elements of the scene measured by number of people per time interval.
2. To globally integrate state of the art pedestrian detection algorithms results per view. To this aim, we need to:
   (a) Create a pedestrian detector fusion mechanism to take advantage of the multi-camera scenario sharing detections from one camera to the others.
   (b) Increase pedestrian detection algorithms performance by the use of semantic constraining information to suppress false detections.

1.3 Thesis Structure

The master thesis is divided into the following chapters:

- Chapter 1. Introduction.
- Chapter 2. State of the Art.
- Chapter 4. Developed Application.
- Chapter 3. Proposed System.
- Chapter 5. Results.
- Chapter 6. Conclusions and Future Work.
- Appendices
- Bibliography.

Chapter 2

State of the Art

As explained in Chapter 1, the analysis of a public crowed space embraces many different algorithms from Computer Vision disciplines.

This Chapter aims to study the State Of the Art in pedestrian detection approaches. In addition, it also covers the topic of contextual information and specifically, the algorithms in the field of semantic segmentation. The advantages and disadvantages of analysis in multi camera scenarios are also discussed.

2.1 Pedestrian Detection

Pedestrian detection (PD) has been a hot research topic in Computer Vision during the past few years due to its impact in several Computer Vision applications. Its main objective is to identify a potential object as a person by automatically detecting its position and relative size in the scene.

Nowadays, it can be consider a partially-solved problem. Although there are excellent PD in the literature, there is no algorithm able to effectively perform PD on a generic scenario. This is the main reason why PD is still one of the most researched areas in Computer Vision.

The complexity related to PD lies on the large amount of available data-sets with different video and people characteristics including challenges such as: people occlusions, poses and scales and scenes captured under extreme illumination conditions. Caltech [12] −recorded on a vehicle in an urban environment−, ETHZ [13] −recorded from a chariot which moves through pedestrian paths−, TUD [14] −static camera in a crossing campus scene− or INRIA [15] −collects precise people images both static and moving− are some of the PD data-sets that have been proposed through the years to train PD. Figure 2.1 gathers some examples of the images in these data-sets.

(a) Caltech [12] (b) ETHZ [13]

(c) TUD [14] (d) INRIA [15]

Figure 2.1: Examples images of PD data-sets

2.1.1 Classical Pedestrian Detection Approaches

Several PD in this State of the Art are arranged under three different topics. First, different pedestrian descriptor schemes are explained. Second, person detection approaches are studied. Finally, various approaches to define person model are analyzed.

Person Description

An organization of existing PD approaches based on the descriptor may start with the Histograms of Oriented Gradients (HOG), which in combination with linear Support Vectors Machine (SVM) have been mainly used to describe pedestrian shape [15]. Differentially, discriminative Part Models (DPM) such as [16] propose to divide the human body into different parts (head, trunk, legs...) and search for their combination on the image to extract PD.

Others PD approaches are based on the use of the Aggregate Channel Features [17]. Algorithms based on ACF rely on a combination of different channels such as normalized gradient magnitude, histogram of oriented gradients (6 channels), and LUV color channels to achieve the final detection.

Person Detection

In [18] object detection is defined as the extraction of potential object candidates to be a person from a scene. Mainly object detections algorithms are:

- Sliding window −also known as− exhaustive search: uses an efficient classifier to test every possible image window. Parameters such as window size, or overlapping between them, are common tuning values that increase or decrease the performance of the detector. These methods usually need from 10^4 to 10^5 windows per image to perform decently. This number grows exponentially for multi-scale detection. If the complexity of the core classifier is increased in every window testing, the computational time will end up being not affordable.

- Segmentation: Uses segmentation as a preliminary step for PD. The use of algorithms such as background subtraction lead to the segregation of the image. Alternatively, color segmentation based on color skin detection can be used to restrict people search. By all means, segmentation severally reduces person candidates reducing the computational time.

- Segmentation + Exhaustive search: Alternatively, a combination of previous techniques. In this case the previous step of segmentation does not lead to final candidates but to a delimited small area that could contain some candidates objects. After the segmentation process, a sliding window technique is performed over the reduced scene area. In this case, improvements from both approaches are exploited as the computational cost of the exhaustive search (which is its main drawback) is reduced by the use of segmentation. Figure 2.2 depicts a flowchart for a generic PD approach which relies on the combination of segmentation and exhaustive search.

Person Model

The model of a person can be considered as the set of characteristics used to discriminate between people and any other object in the scene. In [18] three different types of person model are assumed:

- Based on appearance: Most of the available detectors in the State of the Art use appearance information to define the person model. In this group one can differentiate two approaches to describe the shape of a person.

  ✧ Holistic: People are defined as a unique and indivisible region or shape.

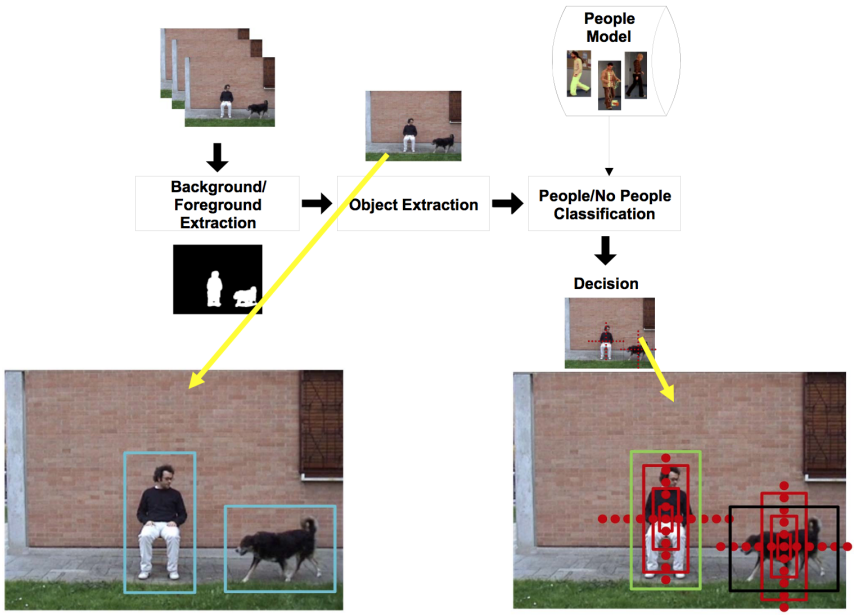


Figure 2.2: Generic Pedestrian Detector Example Diagram. Image extracted from [18]

✧ Part-based: Rely on more complex characteristics where a person is defined as a combination of multiple shapes, regions or parts of its body.

Examples of appearance-based detectors are those that use silhouettes to classify people, either from an holistic or a part-based basis, or color distribution.

- Based on motion: Human appearance is likely to change due to environmental factors such as light conditions, clothes or camera settings. In addition, people variability in terms of height, weigh and poses make appearance likely to vary. For these reasons, some approaches try to get rid of these factors and detect pedestrians using only its motion information. For instance in [19], detections are based on a periodic motion analysis.

- Based on appearance + motion: Algorithms such as [20, 21] merge both appearance and motion information. Most of these algorithms combine people detection and tracking, targeting to improve people tracking rather than PD.

In [22] a comparison of the performance of PD on different data-sets is made. This comparison is partially included in Table 2.1. See VPU Website[1] for the complete table.

[1] http://www-vpu.eps.uam.es/DS/PDbm/results.html

Video	HOG	ISM	Fusion	Edge	DTDP	ACF	Faster-RCNN
1	89.3	71.4	34.9	84.9	96.7	99.3	**99.7**
2	69.2	82.9	92.5	90.2	77.1	77.1	**98.2**
3	55.6	75.7	64.3	71.7	68.9	68.9	**82.9**
4	10.1	1.0	0.5	5.4	33.9	33.9	**37.5**
Average AUC	56	57.5	48	63	69.1	69.8	**79.5**

Table 2.1: Pedestrian Detection Performance. Adapted from [22] (selected approaches). Metric for this evaluation is the average AUC.

2.1.2 Recent Trends in Pedestrian Detection

During this section recent PD trends in terms of person detection and description are presented.

Person Detection

PD based on HOG, DPM and ACF generally rely on "sliding window" detectors , however, as mentioned before one of the main drawbacks of this approach is the high computational time needed to achieve good performance.

One of the most successful solutions to overcome this time consumption problem without losing detection quality is the use of object proposals [6].

Object proposals approaches perform a complete search over an image to detect potential object candidates. These candidates are detected as image areas with visual properties that distinguish them from the scene background.

In general, object proposal approaches reduce pedestrian candidates with respect to sliding-window like algorithms and generally outperform segmentation based methods. This advantages lead to a higher object recall and more efficient detection processes. Successful examples of using proposals to improve and speed-up detection include Faster R-CNN [5].

In [6] three set of proposal methods are analyzed:

- **Grouping proposal methods** attempt to generate multiple, and so, overlapping segments that are likely to correspond to objects. Here one can distinguish between three types of methods according to how they generate proposals. Methods can generate proposals by groping super-pixels (SP), solving multiple graph cut (GC) problems or finally, using edge contours (EC). Among those

that use SP we can find Selective Search [23], Randomized Prim's [24], Rantalankia [25] or Chang [26]. Those that use GC are CPMC [27], Endres [28] or Rigor [29]. Finally Geodesic [30] and MCG [31] use EC to obtain proposals.

- **Window scoring proposal** methods are an alternative approach to score each candidate window according to the probability to contain an object. Usually this methods tend to be faster and, in addition, they typically extract only bounding boxes. One can find among other approaches Objectness [32], Rahtu [33], Bing [34], EdgeBoxes [35], Feng [36], Zhang [37], RandomizedSeeds [38].
- **Alternative proposal methods**. Apart from the main groups a set of alternate approaches such as ShapeSharing [39] or Multi-box [40] are also used to extract object proposals.

Person Description

In recent years new schemes of PD have been proposed. Detectors based on deep Convolutional Neural Networks (CNNs) have notably improved the accuracy of all the previous analyzed algorithms.

Examples such as ImageNet [41] for image classification, CompACT [42], Fast-RCN [43] or Faster R-CNN [5] for object detection expose that deep convolutional networks usually improve the performance of aforementioned approaches. This fact is clearly presented Table 2.1 where Faster R-CNN outperforms every other approach.

2.1.3 Next Steps Towards Generic Pedestrian Detection

PD is constantly in development and so, some future work lines can be set. In [3] some research directions are proposed that could be of interest in the scope of this work.

1. Use of context information. Starting from the hypothesis that a person is standing on the floor, the ground plane assumption can reduce errors if the detection for both the person and the floor are accurate. This could be achieved by extracting useful contextual information from the scene. In [18] contextual information is added to PD to increase performance. This is one of the main objectives of the work.
2. Occlusion treatment. Usually pedestrians, due to other scene elements such as columns or even other pedestrians appear occluded. When this happens PD performance is substantially degraded under even mild occlusions. Improvements in this area could increase the overall performance of PD in generic scenarios.

2.2 Contextual Information

One can describe contextual information as the set of additional circumstances or facts that can be extracted from a scene besides the target of analysis. Generally, this set of circumstances is a source of information that is not extracted by machine applications but constitutes key evidences for humans, which acquired this knowledge during their life. By just taking a look to an outdoor image a human can derive where the sky will be, what the weather conditions are or which time of the day is. Also, by knowing the place where a video was recorded, one would imagine which objects are more or less probable to be in the scene.

Dealing with computer vision disciplines, contextual information sources also include camera information (such as position, configuration, distance to an object and camera motion), the set of objects that one could detect in the scene and the number of available cameras.

One can divide contextual information into two different levels: global and local . Besides, we can also divide contextual information into two different categories: offline, and online. Finally, we focus on a set of specific methods to extract context −semantic segmentation–.

2.2.1 Global Context

Global context considers descriptions from an image as a whole. For instance, if the context of a scene is known −kitchen−, we can use this information to search for typical objects in this context −e.g. a stove−.

This kind of approaches are focused on psychology studies that suggests that human perceptual processes work following a hierarchically organized process [44, 45]. Our perception system goes from a global structure towards a more detailed analysis in a top-down scene interpretation.

Global context approaches aim to define a scene as an extra source of global information. The structure of a determinate scene image can be estimated by means of global image features as in [46].

2.2.2 Local Context

On the contrary, local context refers to contextual information related to a specific object, e.g. a kitchen table may help to predict the presence of a spoon.

The impact area of an object −in contextual terms− is defined as a set of neighboring objects, patches or even pixels. Algorithms dealing with the extraction of local

contextual information aim to correctly define the area that surrounds an object to precisely detect other object instances.

In [47] the inclusion of local contextual regions such as facial bounding contour are used to improve face detection performance.

2.2.3 Offline

Offline contextual information is defined as the set of circumstances that are computed before starting any kind of analysis procedure. This information leads to external image information that may be used to constraint analysis algorithms. Approaches such as [18] use previously introduced contextual information to improve part-based PD over a scene.

2.2.4 Online

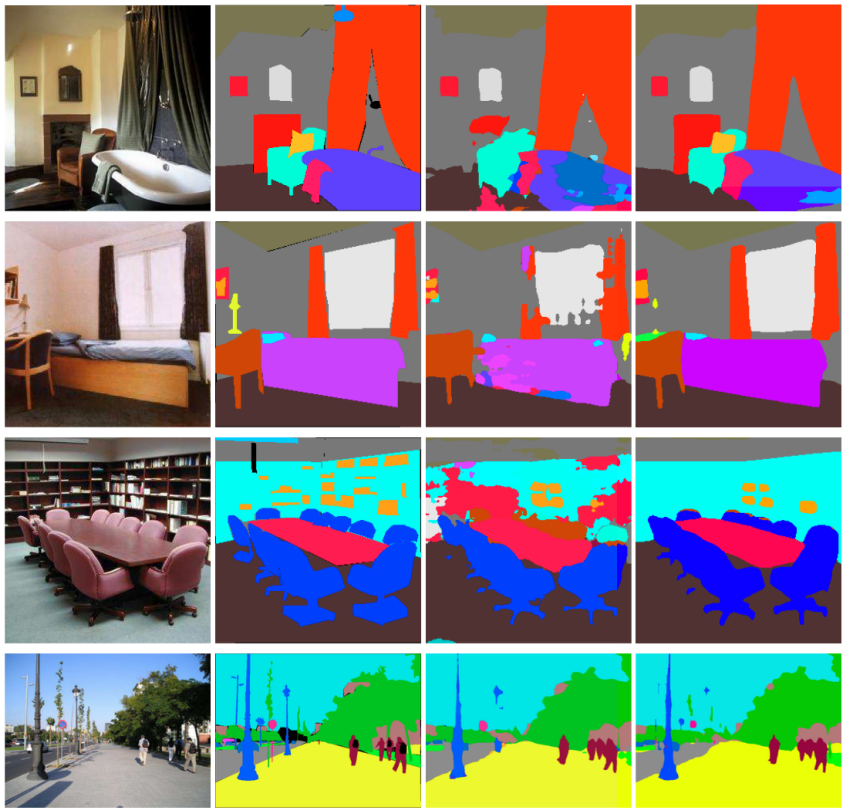
Online information, on the other hand, is to be extracted with the analysis. Online extraction entails a degradation of an algorithm efficiency, albeit allows to dynamically update the context.

Examples of online (and local) contextual information extractors are the algorithms in the semantic segmentation branch. Next section (2.2.5) discuss some of the approaches in this vein.

2.2.5 Semantic Segmentation

Semantic information is defined as the set of high-level elements from contextual information. This can often lead to characteristics such as global image color [48]. It can also describe an image by the position and status of relevant objects [49]. Or it can also be used to define specific image areas in a scene such as walls, corridors or walking paths [7].

Semantic segmentation targets to assign each image pixel a high-level label. If accurately performed, it provides fully semantic understanding −which in terms of Computer Vision−, means that the location of an object within an image is known. A potential result of a semantic segmentation method is depicted in Figure 2.3. In the Figure four different scenes are analyzed and divided into non-overlapping semantic areas.



(a) Image (b) Ground Truth (c) Baseline (d) PSPNet

Figure 2.3: Semantic segmentation on ADE20K data-set [50] by the algorithm described in [7].

Semantic segmentation is a recent trend and nowadays, and so, it remains a significant challenge for the computer vision community. Due to its short-life term there is not yet a complete survey available in which algorithms are deeply analyzed and compared. However, there are a set of benchmark where developers can upload their obtained results with a given dataset and so, algorithms performance can be compared.

An example of a popular benchmark is the Cityscapes Data-set [51]. It is a large-scale data-set that contains stereo video sequences recorded in street scenes from among 50 different cities around the world. This data-set presents categories annotations over pixels in more than 5000 frames. The set of categories is presented in Table 2.2, whereas a subset of the compared methods in the Cityscapes Website[2] are depict in Table 2.3. Only those algorithms that have more than 80% on IoU class metric have been included in the Table.

[2] https://www.cityscapes-dataset.com/

Category	Classes
Flat	road · sidewalk · parking · rail track
Human	person · rider
Vehicle	car · truck · bus · on rails · motorcycle · bicycle · caravan · trailer
Construction	building · wall · fence · guard rail · bridge · tunnel
Object	pole · pole group · traffic sign · traffic light
Nature	vegetation · terrain
Sky	sky
Void	ground · dynamic · static

Table 2.2: Cityscapes Data-set Class Definitions

Algorithm Name	IoU Category	IoU Class	Available Code
motovis(Anonymous)	**91.5**	**81.3**	No
PSPNet [7]	91.2	81.2	**Yes**
ResNet-38 [52]	91.0	80.6	**Yes**
NetWarp (Anonymous)	91.0	80.5	No
TuSimple_Coarse [53]	90.7	80.5	**Yes**

Table 2.3: Cityscapes Data-set Challenge Results (Intersection over Union metric)

PSPNet [7], ResNet-38 [52] and TuSimple_Coarse [53] are all based on convolutional networks. This fact reveals that deep convolutional neural networks have led to significant improvement over previous semantic segmentation systems since the presentation of AlexNet [41] in 2012.

However, even when using CNNs, the main difficulty of scene parsing is related to the type of scene and to label variety.

TuSimple_Coarse [53] propose a combination between dense upsampling convolution (DUC) to generate pixel-level prediction and a hybrid dilated convolution (HDC) framework.

ResNet-38 [52] on the other hand, propose not only to not increase CNNs depth, but rather to ensemble many relatively shallow networks to increase performance. Their approach also improves usability reducing memory use and sometimes even training time.

Finally, PSPNet [7] deals with this problems assigning relationships between different labels, i.e. an airplane is likely to be in runway or flying in the sky while not over a road or in the water. This relationships reduce slightly the complexity of having large amounts of labels to predict and improve the general performance of the algorithm.

In Figure 2.4 some visual examples of how this algorithms perform on Cityscapes Data-set frames are displayed.

2.3 Multi-camera scenarios

The use of multiple cameras is a common setup when dealing with video surveillance problems. One can define a multi-camera scenario as a space that has more than one video camera recording. Ideally, the recordings for the different cameras are temporally aligned −synchronized−. Having N camera instances allows to observe the same event or object of interest from different points of view. This leads to a set of advantages in the scope of our work when dealing with PD and semantic classification and also, to some unavoidable disadvantages.

- Advantages

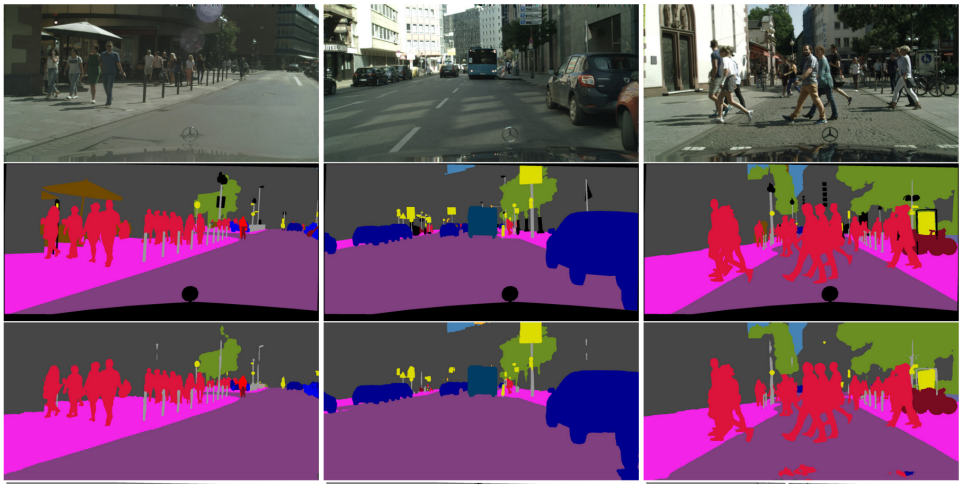
  As discussed in Section 2.1 one of the research paths towards PD is generic occlusion handling. In this case, the use of a multi-camera scenario with relating camera views could help. This could be achieved by reprojecting detections from one camera to another whose miss rates are high as in [54].

  When dealing with semantic segmentation, the inclusion of a multi-camera system may arise some benefits. A single-camera system could lead to misclassification of labels in the image. In a multi-camera system one camera instance may help to refine the classes in another one provided that, evidentially, the views of both cameras partially overlap [55].

- Disadvantages

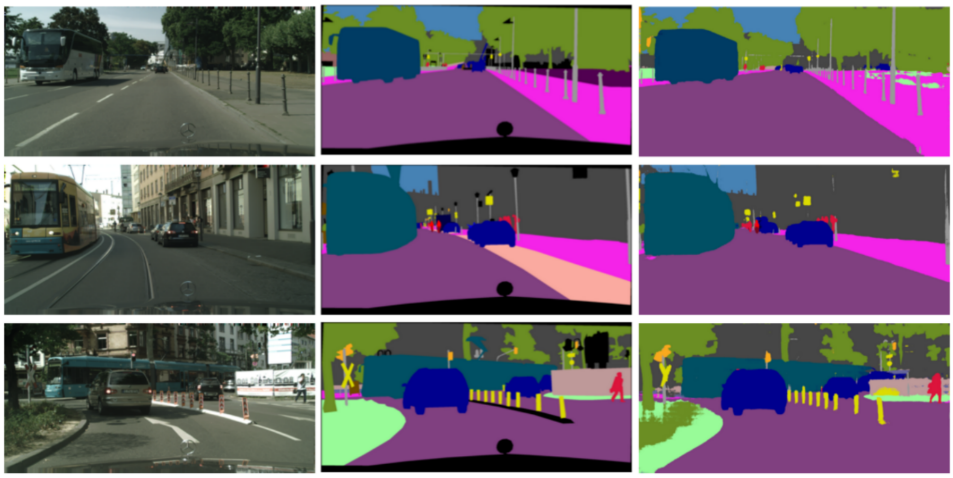
  The main disadvantage when dealing with multi-camera systems is the exponential grow of computational time. Algorithms should be performed N times. This issue could be solved by the use of parallel coding to process cameras views simultaneously.

  Besides, the use of multiple cameras entail two additional tasks: the temporal alignment of the views −synchronization− and the spatial arrangement of the different views −homographies−.



(a) ResNet-38 Algorithm. From top to bottom: Input images, ground truth and results.

(b) PSPNet Algorithm. From left to right: Input images, ground truth and results.



(c) TuSimple_Coarse Algorithm. From left to right: Input images, ground truth and results.

Figure 2.4: Semantic Segmentation result examples on Cityscapes Data-set

Chapter 3

Proposed System

During this Chapter our proposed system is analyzed. We start from the contextual model generation and pedestrian detectors. Then, the fusion of all the obtained elements in a multi camera system and semantic filtering are described. Finally, a case of example which generates statistical usage data from semantic areas is proposed. Figure 3.1 depicts the flowchart of the modular proposed method.

3.1 Contextual Model Generation

One of the main objectives in the scope of this work is to perform semantic segmentation, which −as explained in Chapter 2− targets to divide one frame into different semantic areas. The relative position in the scene for elements such as doors, walls, paths, and columns is required to achieve further objectives such as multi camera pedestrian constraint and statistical data extraction.

For this complex task the algorithm PSP-Net [7] presented in Chapter 2 is used. We choose PSP-Net because at the moment of this work was the one with available code achieving the best results (see Table 2.3). The goal of this algorithm is to assign each pixel in the image a category label.

3.1.1 Pyramid Scene Parsing Network

It uses a deep Convolutional Neural Network (CNN) called Pyramid Scene Parsing Network (PSPNet). This network is designed to improve performance for open-vocabulary object identification in complex scene parsing. The structure of the network is represented in Figure 3.2.

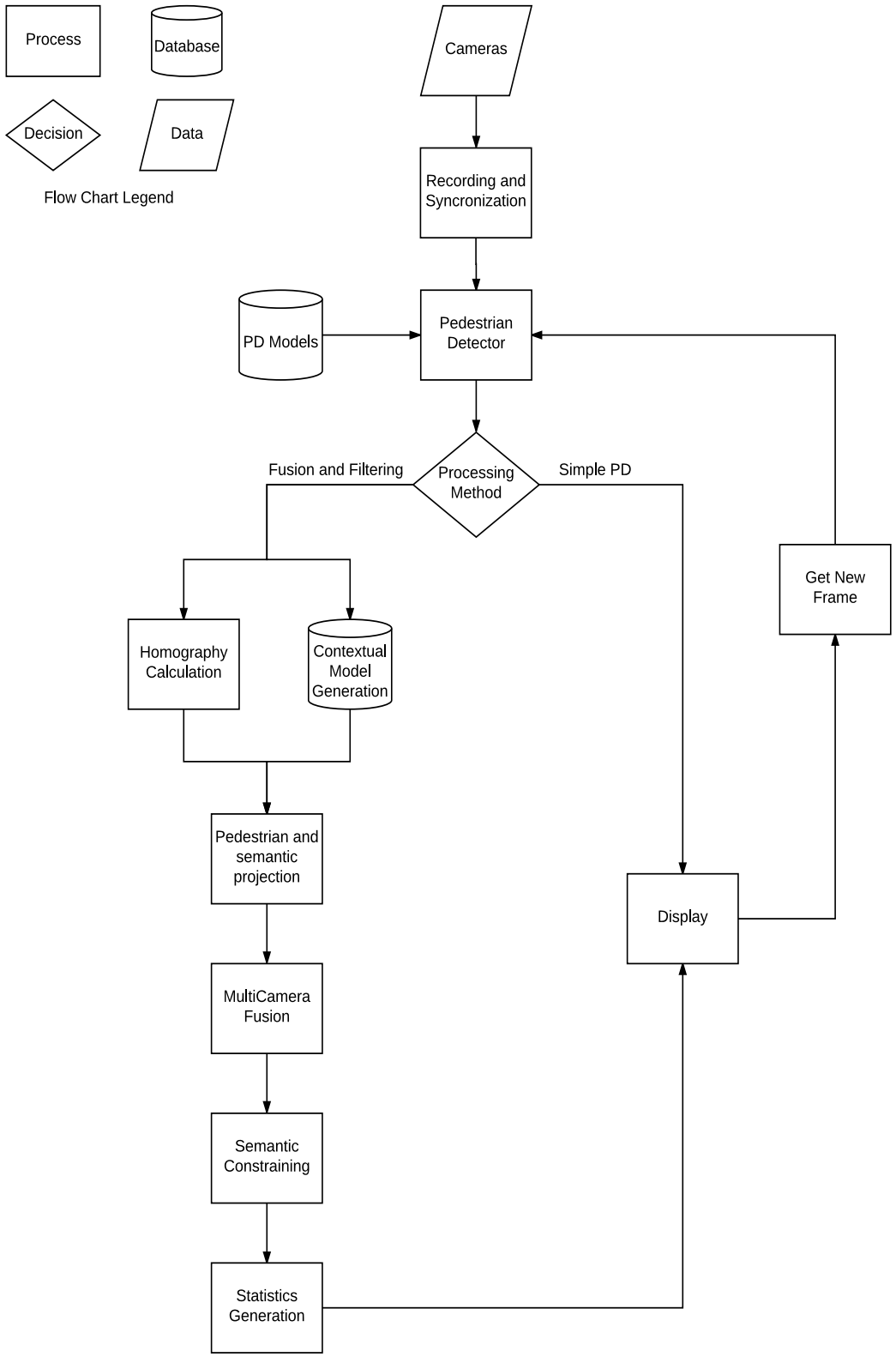


Figure 3.1: Flowchart of the proposed method

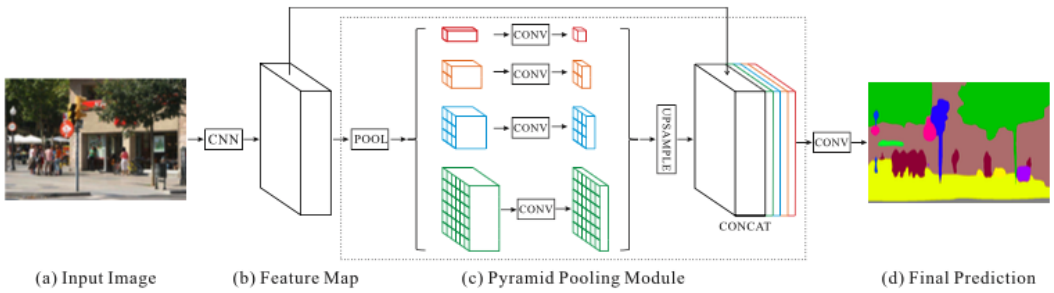


Figure 3.2: Overview of the proposed PSPNet

Algorithm Stages

The algorithm is divided into different stages:

1. Image in 3.2.(a) is processed it through a pre trained CNN called ResNet [56]. The objective is to get the full feature map 3.2.(b) of the last convolutional layer. The final feature map in this step is 1/8 of the size of the input image.

2. Apply the main contribution of [7], called the Pyramid Pooling Module 3.2.(c). The main objective of the module is to collect a few levels of information, much more representative than global pooling. It separates the feature map into different sub-regions and forms pooled representations for different locations. Here a set of pooling, convolutional and upsampling layers are applied to harvest different sub-region representation in N different scales.

3. Concatenation layers are used to form the final feature representation by fusing the feature map extracted in 3.2.(b) and the Pyramid Pooling Module output. This final feature carries both local and global context information.

4. The representation is fed into a convolutional layer which gets the final per-pixel prediction 3.2.(d).

Semantic Segmentation Particularization

PSP-Net[1] comes with a set of three different pre trained Caffe[2] models for three different datasets. The main difference between the models for our scope is the environment in which the network has been trained.

[1] https://github.com/hszhao/PSPNet
[2] http://caffe.berkeleyvision.org/

- ADE20K: This dataset is the most challenging as it has up to 150 different labels in a wide range of scenes. The scenes go from interior room places to outdoor scenarios.

- VOC2012: It contains 20 object categories and one background class from diverse indoor and outdoor scenes.

- CityScapes: The last dataset defines 19 categories containing both stuff and objects. All the available sequences have been recorded from a driving car while driving in the street.

Model Particularization

As one can observe the three different models represent different object categories in different real spaces. In our case we select the model based on two main reasons:

1. The model should have been trained with indoor scenes. This leads to discard those models that represent only outdoors scenes as we would like our approach to be used in an interior scenario. This excludes CityScapes dataset from our options as all the classes and sequences used for training are from outdoor scenes.

2. From the trained indoor models we have to choose between those whose categories best fit in our work. In this case VOC2012 dataset uses classes such as boat, airplane or table which are not interesting for our segmentation problem and it does not have classes such as door or wall which are really important for us.

Selected Model

Considering these two reasons, we have selected the model ADE20K because as said, it has elements such as walls, floor, person and column in its model.

However, we consider that most of the 150 label categories may be unused in our procedure, so, the number of classes from the model has been reduced to the 21 classes of our interest. Position and scores for those objects are the only ones obtained. This class limitation leads above all in a considerably hard drive space saving. In Table 3.1 the final 21 selected classes are exposed.

wall	building	floor	ceiling	road
window pane	person	door	table	chair
seat	desk	lamp	column	counter
path	stairs	screen door	stairway	toilet
poster	bag			

Table 3.1: Final classes list from ADE20K to use in PSP-Net

3.2 Pedestrian Detection

Along this section pedestrian State of the Art detectors that have been integrated in the proposed system are presented. Some of them have been chosen due to their efficiency, whereas some have been chosen due to its contrasted good performance (see Chapter 2).

Besides, algorithm source code of all the chosen approaches is available.

3.2.1 Histogram of Oriented Gradients

Histogram of Oriented Gradients, i.e. HOG, is one of the main used detectors along pedestrian detection field. This fact is due to it's extremely simplicity in terms of the descriptor complexity.

Person Descriptor

Pedestrians are described as set of HOG. This means that its shape and appearance can be described by a set of gradients and intensities organized as orientation histograms. These histograms describe intensity distributions from local gradients or border directions. This descriptor can be observed graphically in Figure 3.3.

Model Generation

Once HOG have been used to describe the person shape, Support Vector Machines are used to train a person model and to classify potential candidates as people. SVM are a data classification method formed by a set of supervised training. The aim of this kind of approaches is to produce a model which is able to predict classification labels on a test set based only on the descriptors of the set and the model.

The idea behind SVM is that training vectors are mapped on to a bigger dimensional space in which the data separation, by means of one or many hyperplane, is

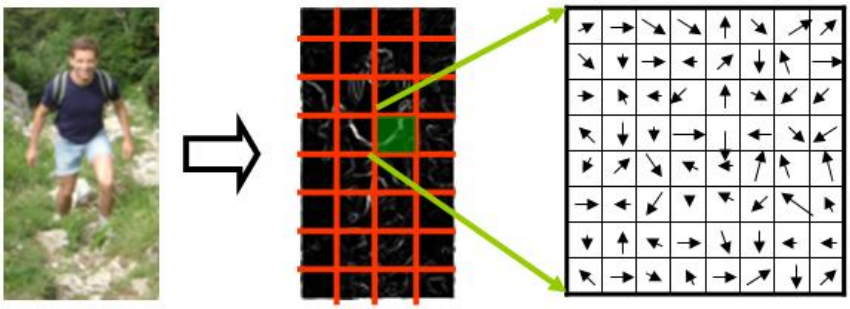


Figure 3.3: HOG Pedestrian Descriptor

much easier to divide than in the original dimensional space.

The combination between HOG and SVM leads to a fast detector that depending on the situation performs decently, although it has some main drawbacks as its lack of occlusion treatment.

The main implementation of Histogram of Oriented Gradient Pedestrian Detector is in OpenCV library for C++.

3.2.2 Deformable Part Model

As was mentioned in the previous paragraph one of the main drawbacks when working with the simple HOG pedestrian detector is that it describes the person model as a whole which leads, inevitably, to the mentioned occlusion drawbacks.

Person Descriptor

Deformable Part Model tries to solve this problem, among others, by defining the model as first, a global coarse template, secondly, several higher resolution part templates and finally a spatial model for the location of each body part. This description can be observed in Figure 3.4.

Model Generation

Both global and part templates are modeled with Histogram Of Gradient features and the model is built by using an improving over SVM called latent SVM. In addition, scores for every detection are obtained by applying a root filter on the window plus the sum over parts of the maximum over placements of the part filter score on the resulting sub window minus the deformation cost.

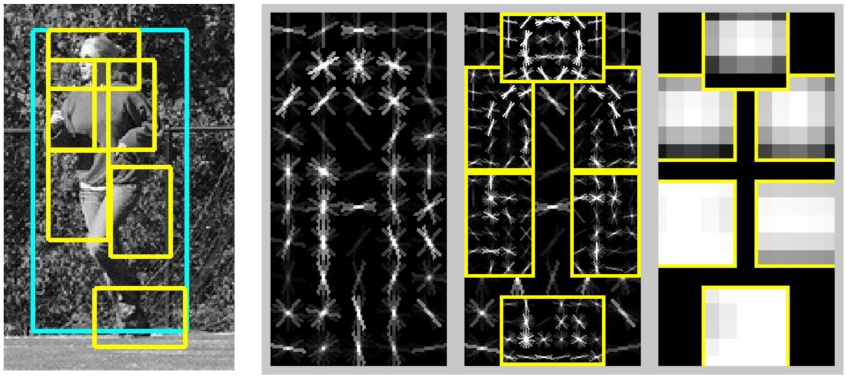


Figure 3.4: Detection example obtained with the DPM person model. From left to right: Original image + detections, global coarse model, part templates and spatial model for each part. Image extracted from [4].

Advantages over alternative PD Approaches

The use of this detector leads to a set of advantages than when using with others simpler PD approaches. In terms of pedestrian occlusion treatment, we are able to detect those people that have been occluded by something in the scene just by detecting some visible part. This outperforms other detectors while working with crowded scenes in which holistic methods have problems that lead for instance to groups of people being detected as a unique detection. DPM DPM is ideally able to separate them onto different person instances.

However, its scanning window approach as well as the part based model lead to some computational cost that increases the needed time to obtain detections.

The main implementation of Deformable Part Model Pedestrian Detector can be found in OpenCV library for C++.

3.2.3 Aggregated Channel Features Detector

The basic idea behind ACF detector is to increase other approaches performance by the use of many different channels to describe an input image I.

Algorithm Stages

In Figure 3.5 one can observe the working path of the mentioned detector.

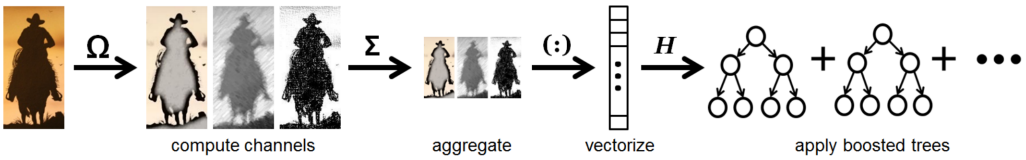


Figure 3.5: Aggregated Channel Features architecture. [17]

1. Given an initial image I, several channels are computed. This channels are named $C = \Omega(I)$. In [17] a set of 10 channels are used to achieve state of the art performance in pedestrian detection:

   (a) Normalized gradient magnitude (1 channel).

   (b) Histogram of oriented gradients (6 channels).

   (c) LUV color channels (3 channels).

2. After the computation, every block of pixels in C is summed and the resulting lower resolution channels are smoothed.

3. After a vectorizing process, features are single pixel lookups in the aggregated channels. Boosting trees are then used to learn this features (pixels) in order to distinguish people from the background.

3.2.4 Fast Region-Based Convolutional Network

Fast Region-Based Convolutional Network (Fast R-CNN) is used to perform offline pedestrian detection in our proposed system.

In 2.1.2 we mentioned that Fast-RCNN must use objects proposals for its usage. In our developed system MCG [31] grouping method is used.

Algorithm Improvements over R-CNN

Fast-RCNN method sets its contributions in a several number of innovations to improve training and testing speed over its fundamental base R-CNN:

1. Higher detection quality (mAP).

2. Training is single-stage, using a multi-task loss.

3. Training can update all network layers.

4. No disk storage is required for feature caching.

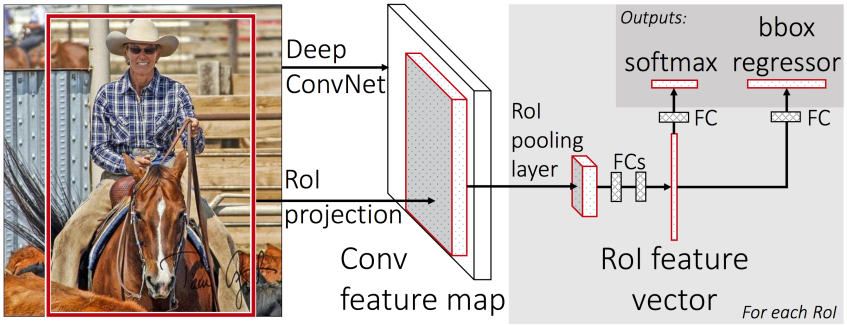


Figure 3.6: Fast-RCNN architecture. [43]

Algorithm Architecture

In Figure 3.6 one can observe the general Fast-RCNN architecture. The input for the Fast-RCNN network are the entire desired image that one wants to process and the set of object proposals. Algorithm stages are:

1. Process the whole image with several convolutional and max pooling layers to produce a convolutional feature map.

2. For every object proposal present in the image, a region of interest (ROI) pooling layer extracts a fixed-length feature vector from the feature map.

3. Each feature vector is after, introduced onto a sequence of fully connected (FC) layers that finally diverge into to output layers.

   The first one produces probabilities estimates over K object classes.

   The second one outputs four real numbers for each of the K object classes. This set of 4 values encodes the final bounding box positions for one of the objects from the K classes.

In this case, both Fast-RCNN detector and MCG object proposal extract are implemented within external Matlab libraries and so, should be computed offline and introduced externally to the application.

3.2.5 PSP-Net

As we explained in Section 3.1 PSP-Net has been trained to detect people as a class. In our proposed system we also use this approach by encapsulating connected components and pixels labeled as pedestrians. Similarly to Fast-RCNN, PSP-Net results are computed offline and introduced externally to the application.

3.3 Multi-Camera Fusion

Once the semantic model and pedestrian detections are obtained the next step is to combine the information from different camera sources and project it onto a common plane. We called this process multi-camera fusion. During this section multi-camera scenarios, generation of reference planes and pedestrian fusion is deeply explained.

3.3.1 Multi-camera Scenario

Our system runs on a multi-camera scenario, hence, the scene can be observed from different points of view. In Figure 3.7 we can observe how the scene is configured with 3 static video-surveillance cameras. Two of them are placed at the sides of the scene, while one is at the bottom part. The starting views from the three static cameras are displayed in Figure 3.8.

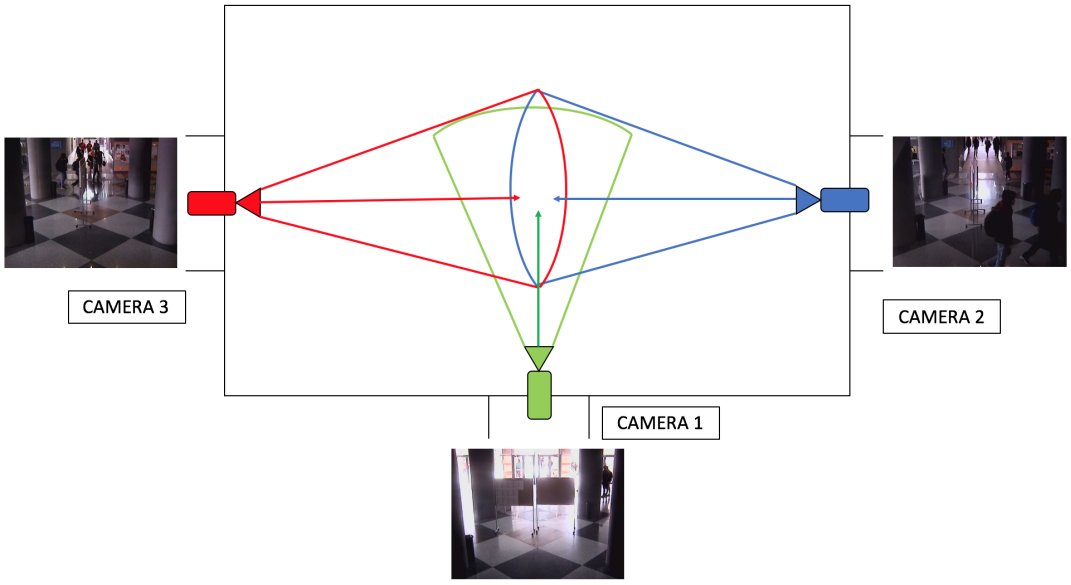


Figure 3.7: Multi-camera configuration

Analyzing a multi-camera scenario has some advantages. Same scene area (as depicted in Figure 3.8) can be observed from different points of view. This implies that for instance, detections from one camera can be used to create new detections in another camera.

(a) Camera 1

(b) Camera 2

(c) Camera 3

Figure 3.8: Initial Camera Views

3.3.1.1 Cenital Plane Homography

Homography calculation is a pivotal task in our work and is the main base for the multi-camera fusion of all the different information.

The objective is to compute an homography matrix ${}^{\pi_{ref}}H_F$ that relates one camera frame F_t at a time t, to a so called cenital plane π_{ref}, i.e. a bird-eye representation of the scene.

With this homography matrix we are able to transform every frame pixel $F_t(x_1,\, y_1)$ to its corresponding position on the cenital plane $\pi_{ref}(x_2,\, y_2)$, i.e. from $2D\,World \rightarrow$ $3\mathrm{D}\,World$. This process is expressed in Eq 3.1.

$$P_i = \lambda \cdot {}^{\pi_{ref}}H_F \cdot p_i \tag{3.1}$$

, where λ is a scale factor, P_i is the subset of points from π_{ref} and p_i is the subset

of points of the F_t.

This homography process, apart from transforming frame F_t perspective, enables to project pedestrians detections to the cenital plane.

In order to compute an homography, a relation between at least 4 points in each of the two images needs to be computed. In our proposed system, these images are a camera frame F_t and a computer-generated cenital view plane π_{ref}. This means that there is not a real correspondence for pixels and so, it is not possible to use a point descriptor algorithm to extract common points between both images. User has to manually select the points that represent the same spatial place in the images using the Graphical User Interface and then the application computes homography matrix ${}^{\pi_{ref}}H_F$.

3.3.1.2 Cenital Plan Design

For the proposed system the image depicted in Figure 3.9 has been used as cenital view plane π_{ref}. The complete creation of the plane is detailed in Appendix A.

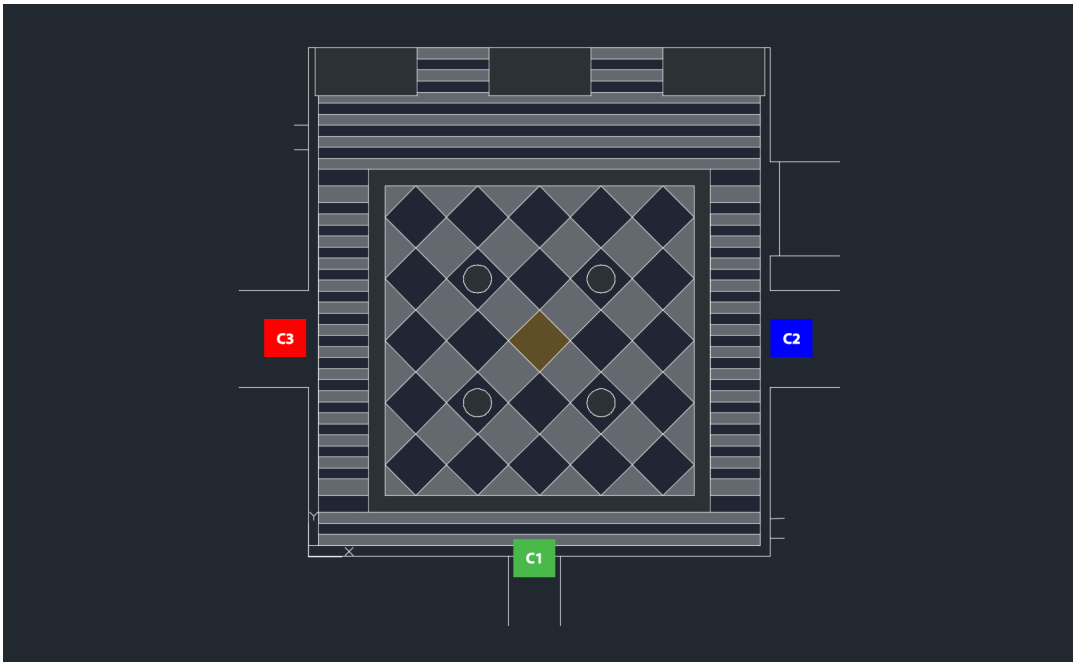


Figure 3.9: Cenital plane with camera positions

This plane has been designed in a highly detailed way -e.g. see the floor tiles– providing enough scene evidences for the homography point selection and estimation process.

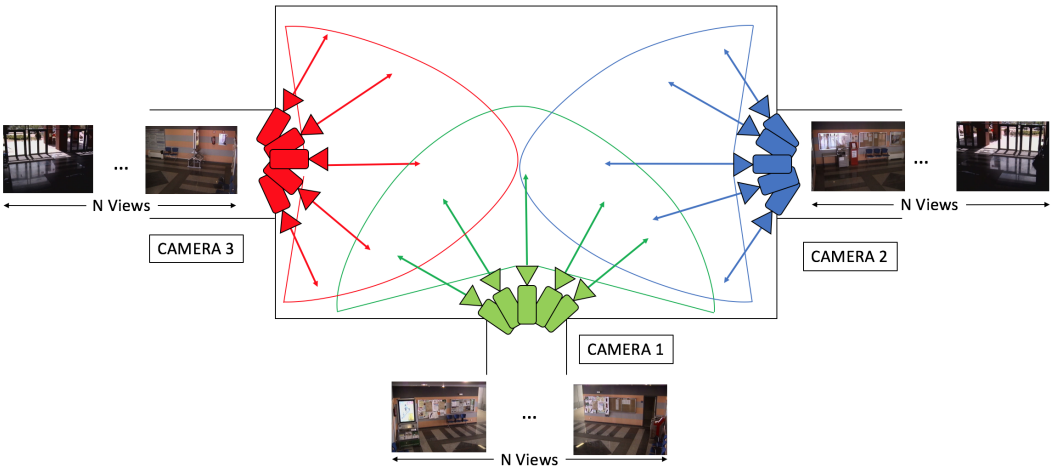


Figure 3.10: Multi-camera configuration with panning setup

3.3.1.3 Camera Panning

If cameras are static and pointing to the center of the scene (Figure 3.8) vision amplitude compared to the complete scenario is quite limited due to the low vision range of the cameras. They are covering the central part leaving completely unattended the lateral parts of the scenario, precluding the analysis of the whole scene.

A solution for the low vision range problem is to include panning movement in the cameras thanks to their PTZ technology. This solution is presented in the scene diagram in Figure 3.10.

3.3.1.4 Camera Panning Discretization

Whereas camera panning allows to widen the visualization area, also entails a operative problem. A specific homography matrix ${}^{\pi_{ref}}H_{Frame_t}$ for each camera position is required for cenital projection, which is an impractical solution.

Instead, we propose to sample the panning tour to select N views from the video sequence. This leads to the obtention of an homography codebook in where an homography matrix ${}^{\pi_{ref}}H_{View}$ for each of the N camera views is computed. This codebook of views and homographies is used for projecting onto the cenital plane the whole video sequence and its detections. Nevertheless, with the codebook, a set of N homographies may be used to project the same frame so one needs to be chosen. This problem is analyzed in the following section.

3.3.1.5 View Selection

One have to choose between the set of N codebook homography matrices to project frames and detections onto the cenital plane. Frame F_t should be compared to each of the N views to obtain an spatial correspondence and so, use the correct codebook homography matrix.

Inter-image Comparison

In order to compare two images a comparison in terms of points of interest matching is performed. To this aim, we use the AKAZE detector and descriptor [57] (Explained in Appendix B).

1. Codebook N views are described in terms of points of interest using AKAZE.
2. Frame F_t points of interest are also extracted.
3. F_t points of interest are compared by Brute Force (BF) to each of the N set of points of interest from codebook views.
4. The comparison that obtains more coincidences is the best spatial correspondent and so, the selected view.

Intermediate Homography

If frame F_t is not spatially positioned as the correspondent view, the selected homography matrix from the codebook does not project the points correctly. This problem can be solved by automatically calculating the homography between both images (Frame F_t and codebook view) using the computed points of interest in the Inter-image Comparison section. This means that now, to project detections from the current frame F_t to the cenital plane π_{ref} two homographies are used:

1. Perspective of the original frame F_t is changed to the perspective of the selected view in the codebook via ${}^{View}H_{Frame}$. This process ensures that the used perspective is the same as the one that was used previously to compute the homography.
2. Finally, the frame and its new perspective $\tilde{F}_t$ are projected to the cenital plane by means of codebook homography of the selected view ${}^{\pi_{ref}}H_{View}$.

This process is done identically for RGB, semantic frames or even pedestrian detections. Figure 3.11 graphically depicts Inter-Image Comparison and Intermediate Homography processes.

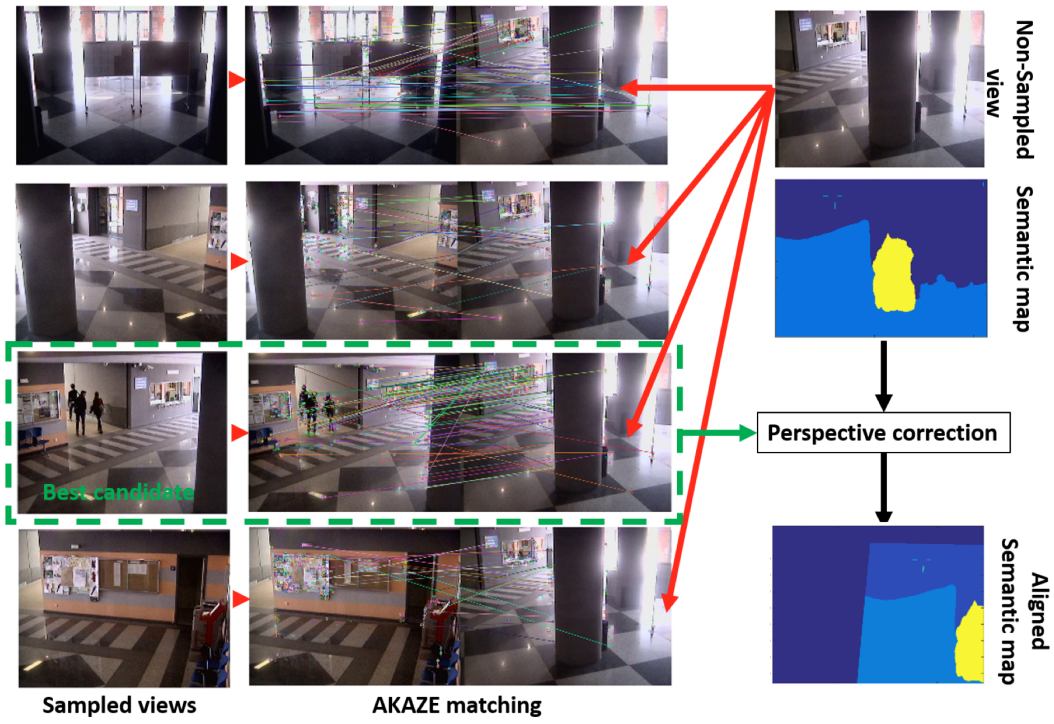


Figure 3.11: View selection process and homography between views computation exemplified by the projection of the semantic map of a non-sampled frame

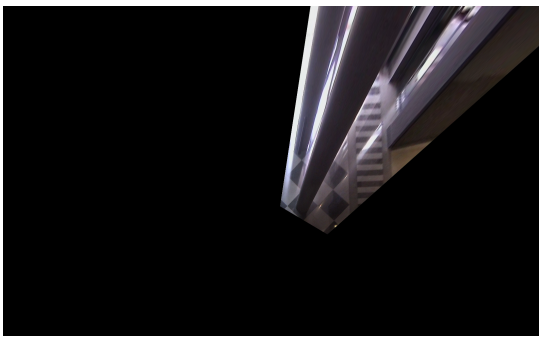
Is essential to say, that, as we have to compare the frame F_t with all of the N views from the codebook, there is a trade off between accuracy and computational time. More sampled views means a better space discretization and more overlapping between them and hence, a lower reprojection error in the Intermediate Homography step. However, the computational time increases exponentially due to comparison between frame F_t and the codebook views. Number of views N has to be selected so it correctly represents the scene space and it keeps the computational time relatively low. In our proposed system we have choose $N = 9$.

3.3.1.6 Video Sequence Synchronization

One of the main issues when dealing with multi camera systems, and specially, those that combine information, is that video sequences should be temporally synchronized. The same frame number F should represent the same exact moment t in all the cameras so combination and fusion of the information is possible.

We have synchronized the processed videos using a called clapperboard technique. This technique aims to synchronize using concrete events that can be observed from all the cameras at the same time. By this, we are able to set a synchronize starting

(a) RGB Projected $Frame_t$



(b) RGB Projected $Frame_{t+\Delta t}$

(c) Semantic Projected $Frame_t$

(d) Semantic Projected $Frame_{t+\Delta t}$

Figure 3.12: RGB and semantic projected frames

point from where the following video frames are temporally aligned.

3.3.2 Reference Planes and Semantic Fusion

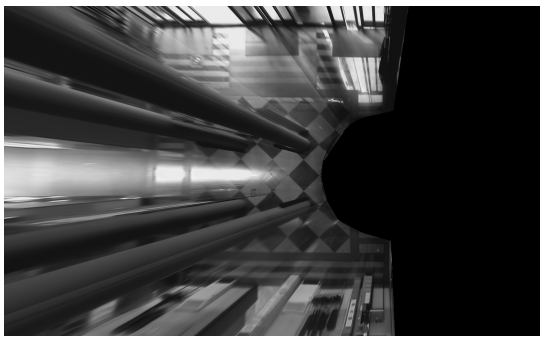
During this Section our proposed method to create RGB and semantic reference planes combining information from the cameras is explained. Besides, the computation of semantic shared areas between cameras is analyzed.

3.3.2.1 Semantic and RGB Reference Plane Generation

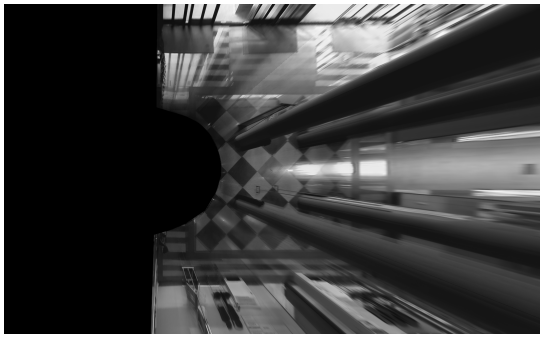
Following the process scheme of Figure 3.11 one can project any video frame onto the cenital perspective. By this, a set of projected RGB and semantic frames for each camera are obtained (Figure 3.12).

Once all the video frames are projected a temporal average is applied through a median filter for each camera. A reference plane π_{ref-C} enclosed in the ground plane is created by temporally fusing all the projected frames from the same camera C. Given a set of pixels $P(x_0, y_0, N)$ from the same spatial position (x_0, y_0) from N different projected images one can define median filter as in Eq 3.2.

(a) Camera 1. ${}^{RGB}\pi_{ref-1}$



(b) Camera 2. ${}^{RGB}\pi_{ref-2}$



(c) Camera 3.${}^{RGB}\pi_{ref-3}$

Figure 3.13: RGB Reference Planes ${}^{RGB}\pi_{ref-C}$.

$$P(\widetilde{x_0},\ \widetilde{y_0}) = \begin{cases} P_{(N+1)/2} & if\ N\ is\ odd \\ \frac{1}{2}(P_{N/2} + P_{1+N/2}) & if\ N\ is\ even \end{cases} \tag{3.2}$$

, where P_N is the N^{th} order statistics of sorted set P.

Obtained reference frames π_{ref-C} from the temporal averaging for RGB can be observed in Figure 3.13 and for semantic information in Figure 3.14.

3.3.2.2 Semantic Fusion in the Reference Plane

Once semantic cenital maps ${}^{Sem}\pi_{ref-C}$ have been extracted, semantic fusion between different camera reference planes can be performed. The main objective is to obtain common semantic areas, i.e. pixels with the same label within two or more cameras. These pixels hypothetically have a higher probability to be that label as they have been equally detected in two cameras. This information may be used to constrain pedestrians detections. As the reference plane is obtained by the homography matrix ${}^{View}H_F \cdot^{\pi_{ref}} H_{View}$, we have only considered floor label pixels to create the common

(a) Camera 1 ${}^{Sem}\pi_{ref-1}$

(b) Camera 2 ${}^{Sem}\pi_{ref-2}$

(c) Camera 3 ${}^{Sem}\pi_{ref-3}$

Figure 3.14: Semantic Median Average ${}^{Sem}\pi_{ref-C}$.

areas. Other labels may have projection errors for not being enclosed in the ground plane. Process to extract shared areas is:

1. Semantic information coming from pairs of cameras is combined to create three common floor semantic areas (Figure 3.15).

2. Using common semantic between pair of cameras one can extract common floor areas for the three cameras at the same time. Those areas have been detected as the same label in three cameras so hypothetically, its probability to be floor class is even higher. Figure 3.16 depicts common floor areas between all the cameras.

Due to the multi camera setup and some scene occlusions the common area shared by all the cameras represents a really reduced area of the room.

3.3.3 Pedestrian Reprojection

During this Section we explain how pedestrian detections are reprojected from its original camera to other frames.

(a) Cam 1 and 2. ${}^{Sem}\pi_{ref-1} \wedge {}^{Sem}\pi_{ref-2}$

(b) Cam 2 and 3. ${}^{Sem}\pi_{ref-2} \wedge {}^{Sem}\pi_{ref-3}$

(c) Cam 3 and 1. ${}^{Sem}\pi_{ref-3} \wedge {}^{Sem}\pi_{ref-1}$

Figure 3.15: Common semantic areas between pair of cameras

Figure 3.16: Common semantic areas for all the cameras ${}^{Sem}\pi_{ref-1} \wedge {}^{Sem}\pi_{ref-2} \wedge {}^{Sem}\pi_{ref-3}$

3.3.3.1 Cylinder Estimation

Detected bounding boxes on one camera frame, due to camera perspective, do not correspond spatially with the exact position of the detected object. This detection error should be corrected before projecting blobs either to another camera instance or to the common cenital plane. In [54] a cylinder estimation technique is proposed to solve this problem. Figure 3.17 graphically represents [54] solution for bounding box transference between cameras.

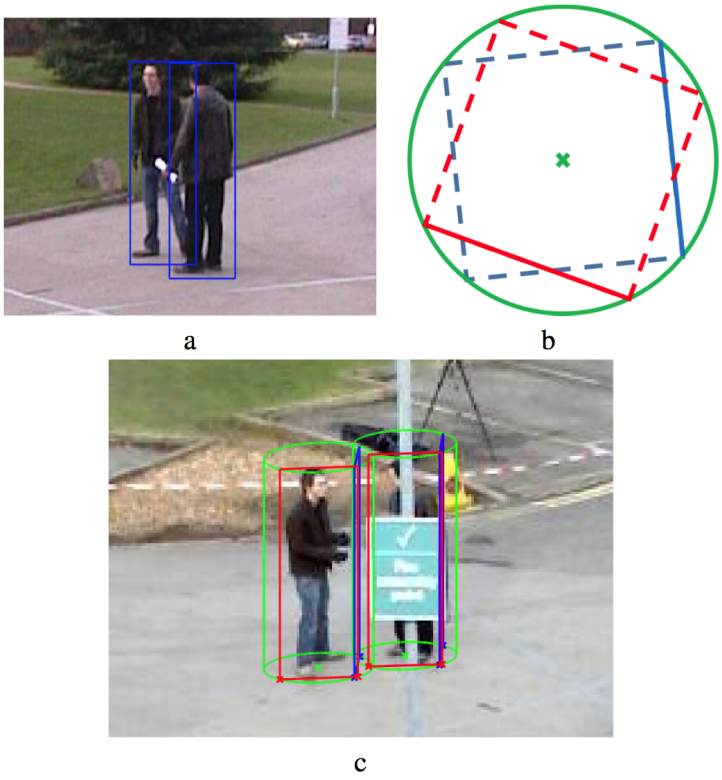


Figure 3.17: Cylinder estimation for camera instance projections. Extracted from [54].

When the blue bounding boxes are projected from image (a) to image (c) they are not correctly placed on the pedestrian. The solution is to compute the cylinder that embraces the square whose side is the blue bounding box (b). Once the cylinder is estimated, the person will ideally be in the middle of the cylinder.

Cylinder estimation has been integrated in the proposed system to correct PD errors. Figure 3.18 represents the method but in this case, applied to the case when the bounding box is projected to the cenital plane π_{ref}. The projection of the bounding box (green line) is not in the same position as the center of the cylinder (end of the

purple line). The center of the cylinder corresponds to a more approximate position of the detected person.

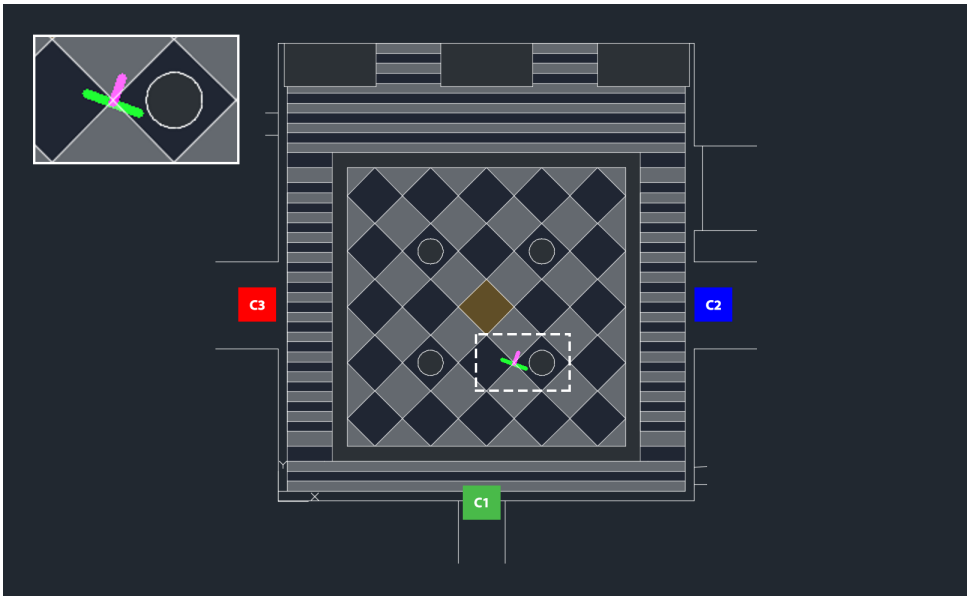


Figure 3.18: Cylinder estimation for cenital view projections

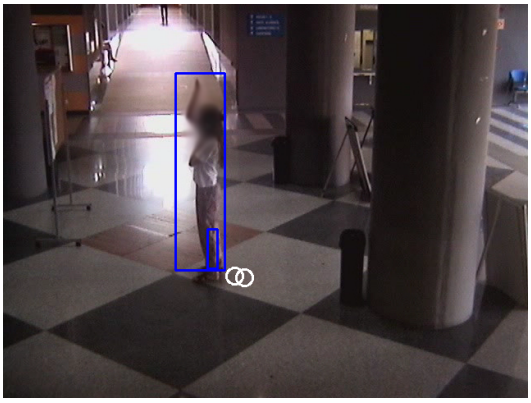
3.3.3.2 Pedestrian Reprojection Between Cameras

When pedestrian detections have been projected onto the cenital plane they can be reprojected from it back to other camera frames, for instance, pedestrian detections from Cameras 2 and 3 may be reprojected onto Camera 1. This process is done when cameras are observing the same spatial area. Ideally, if the three cameras detect the same person, reprojection may not be necessary but, if one of them misses a detection, reprojection method may lead to avoid that miss detection. Steps to fuse detections between cameras are:

1. Detected pedestrians in all the cameras are projected to the cenital plane π_{ref} by the process from Section 3.3.1.5.
2. Projected blobs from one camera are transferred to different cameras as others detector blobs.
3. Reprojected blobs are finally created by the use of a fixed aspect ratio to, using the width of the blob obtain the height (which was lost due to the cenital projection).
4. Non-Maximum Suppression (NMS) is applied in frame F_t to join original detections from its camera and reprojections.

Figure 3.19 represents an example of reprojection. If two white circles are displayed, it means that two bounding boxes from other cameras have been reprojected. However, the absence of a third bounding box means that one reprojection has fused with the original detection while the other is not accurate enough to do it.

(a) Camera 1 + Reprojections from 2 and 3



(b) Camera 2 + Reprojections from 1 and 3

(c) Camera 3 + Reprojections from 1 and 2

Figure 3.19: Pedestrian detection reprojection. Original and reprojected bounding boxes. White circles correspond to reprojected detections from others camera.

3.4 Semantic Filtering

Semantic constraining is used in the proposed system to filter false positives detections in not plausible semantic scene areas. Generally, people are always walking on paths so, floor areas may be used to constrain pedestrian detections to this hypothesis. In our system, common floor areas between pairs of cameras are used instead of shared areas between the three cameras (see Section 3.3.2.2) . This choice is based on the reduced common floor area between the three cameras (see Figure 3.16) in comparison

to the complete scene.

The constraint idea is that a bounding box coming from a camera C_1 is assumed correctly detected if the center of its related cylinder is on the common floor between C_1 and C_2 and also between C_1 and C_3. If that condition is not fulfilled that detection is suppressed.

Figure 3.20 represents different constraining frame examples.

(a) Camera 1

(b) Camera 2

(c) Camera 3

Figure 3.20: Pedestrian semantic constraining. Blobs which projection does not fall on the common floor between two cameras are just displayed by a circle and text label.

3.5 Statistical Usage Data

Along this Section we explain the proposed system to extract statistical usage data frame by frame given a sequence. The aim is to obtain one graph per selected class that measures the amount of people at a moment t of time in the determined semantic

area during the sequence.

3.5.1 Statistical Semantic Map Generation

For this task, rather than creating common areas between cameras as done for the PD constraining, information for all the cameras have been joined without strictly being shared. This have been done to preserve as much semantic information as possible and so, have bigger *floor* or *door* areas than if we were more restrictive.

The result from this step is a unique map that represents the hole spatial area (Figure 3.21).

Figure 3.21: Statistical semantic map

3.5.2 Usage Curves and Paths

We have decided to only consider for this process *floor*, *doors* and *chairs* labels, which are the most accurate detections from the semantic segmentation. Following the same principle as in the previous Section, pedestrians are projected on to the statistical semantic map and so one can be able to know how many people are in each of the designed areas.

In addition, one can divide Figure 3.9 into different subregions of fixed size and

extract in which of them pedestrians density is higher.

3.6 Gaussian Representation of Bounding Boxes

Only for visualization purposes another representation method has been included. In the simple cylinder representation pedestrian are represented with two perpendicular lines, however, this representation lacks of detections score information. To solve this issue we propose a new representation method based on a Gaussian function placed at the middle of the estimated cylinder.

Every pedestrian detection is so, represented as a Gaussian function of the form described in Eq. 3.3.

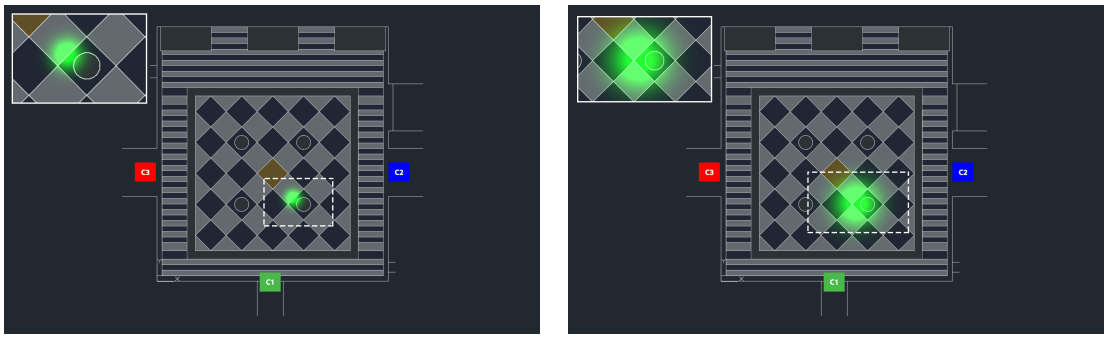
$$f(x,y) = Aexp\left(-\left(\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2}\right)\right) \tag{3.3}$$

In this case A is the amplitude which typically is set to $A = 1$, x_0, y_0 represent the mean of the gaussian which, in our case, is the center of the estimated bounding box cylinder and σ_x, σ_y which is the standard deviation and in our case represents the accuracy of the detection.

In order to relate a detection score S_n with a determinate standard deviation a simple rule explained in Eq 3.4 is proposed.

$$\sigma_x = \sigma_y = (S_n \cdot 10) + 5 \tag{3.4}$$

This Equation has been adapted to the size of the used cenital plane. Detections with higher scores are represented as a narrow gaussian function (Figure 3.22a) while lower scores lead to wide gaussians (Figure 3.22b).



(a) High score detection. Frame 86

(b) Low score detections. Frame 87

Figure 3.22: Gaussian representation examples

Chapter 4

Developed Application

Within this Chapter the developed application in terms of software development is described.

This application is the base for the integration of pedestrian detection algorithms as well as semantic segmentation. Visualization and arrangement of the usage statistics from the different areas of interest is also done by the software.

Application environment should be user-friendly to ensure a correct and easy usage by the end user. It has been developed completely from scratch for the purpose of this Master Thesis.

The application has been developed under QT Creator[1] coding environment in Mac OS Sierra. This decision has fundamentally been based on the following QT characteristics:

1. Its cross-platform characteristic which makes it easily portable from one operating system to another such as Windows or Linux distributions.

2. Its application window designer that allows the programmer to design software windows by using an interface instead of having to create windows by coding (Figure 4.1).

3. The possibility to add OpenCV[2] libraries to the project as well as independent external libraries.

4. Its multi-thread capabilities that enables to perform different code segments in various threads to increase computational speed.

[1] https://www.qt.io/
[2] http://opencv.org/

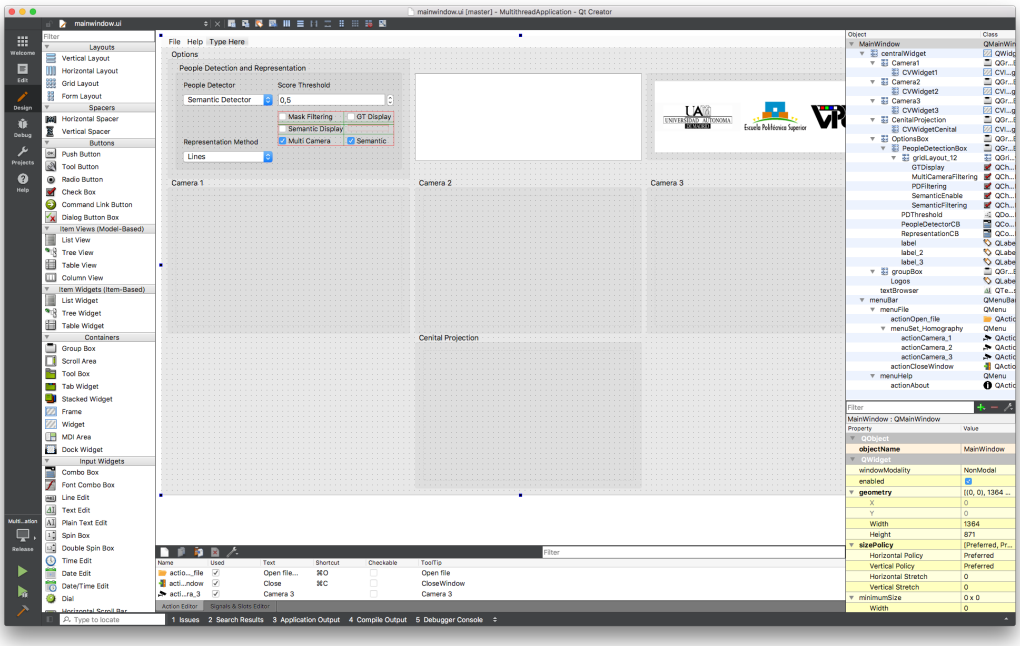


Figure 4.1: QT Main Window Designer

Due to the complexity of some of the algorithms used, in terms of parameter tuning and configuration, two separate applications have been developed: developer and user version.

The first version of the software corresponds to the developer application. It is design so it can be used by programmers or engineers who generally understand concepts of the algorithms running at the backend application. This means that:

1. Variable parameters are available for tuning from the graphical interface.
2. Different pedestrian detectors can be selected.
3. Options and tuning for these algorithms can be done.
4. Results are displayed in different areas.

This version allows to change parameters and methods online. However, this assumes that the user has basic knowledge on how parameters affect the software performance.

For the developer application both single-thread and multi-thread versions have been developed and are discussed on the following lines. All the code for both approaches is available in a GitHub Repository[3].

[3] https://github.com/alexlopezcifuentes/IPCV-MasterThesis/

During the analysis of the application trough the Section some flow-charts are displayed. A common legend for all of them is included in Figure 4.2.

4.1 Single-thread Developer Application

During the first stages of the development and for the sake of simplicity the application has been designed and developed to run under a single thread. This means that all the processing has been done sequentially camera by camera. A simple flow-chart diagram that illustrates the execution path can be seen in Figure 4.3 .

This approach has the advantage that all the code is executed in the same memory segment. This makes really simple, for instance, to share information between cameras. This design is however, only valid if the computational effort is minimum. All the process for the three cameras should be computed one after the other which means that when calculating detections for one camera, the others remain idle. When working with such a multi-camera system with heavy algorithms running −as in the proposed method− the computational time increases exponentially and this design is no longer worthwhile.

4.2 Multi-thread Developer Application

Multi-thread Developer approach can be observed in the flow-chart displayed in Figure 4.4. Now different threads are running in parallel, one for each camera, and so, all the process is no longer done sequentially and computing power of the CPU can be further exploited.

However, as threads are running separately a synchronization strategy should be included to keep consistency in the application.

One thread can process a frame faster than another one due to multiple external reasons, nevertheless, the application should display the same exact frame for all the cameras at the same time. This is specially relevant if results are going to be shared between threads. In our case the synchronization is performed by two barriers −see diagram 4.4−.

- The first one ensures that all the threads have perform PD before sharing these detections to the rest of the threads.
- Second barrier creates a meeting point at the end of the frame processing so a thread waits to the others before sending results to the main display.

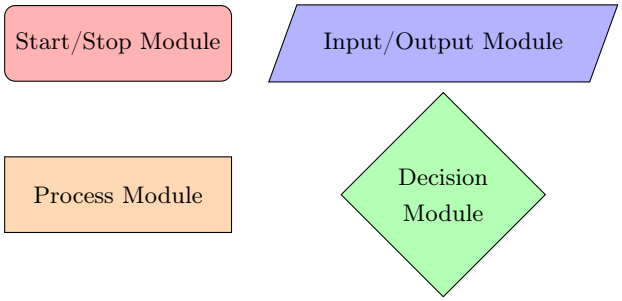


Figure 4.2: Flow-chart legend

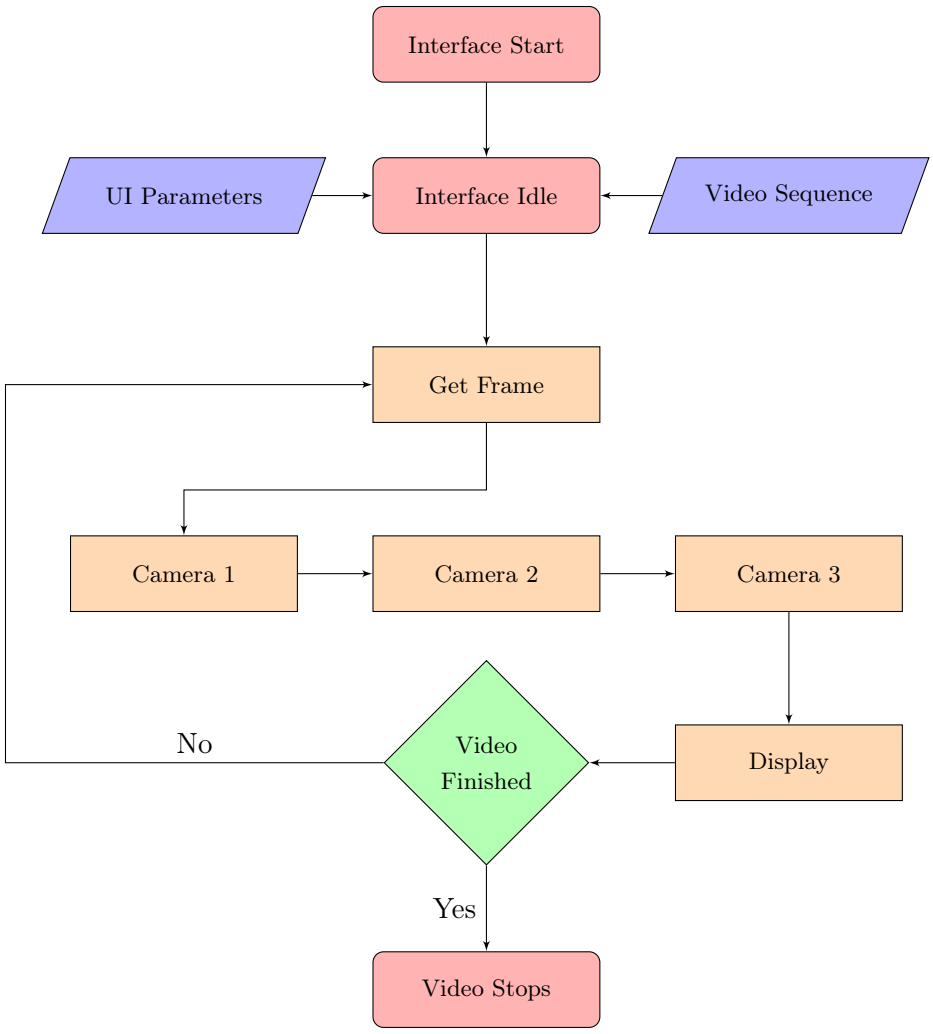


Figure 4.3: Flow-chart diagram for the single-thread application

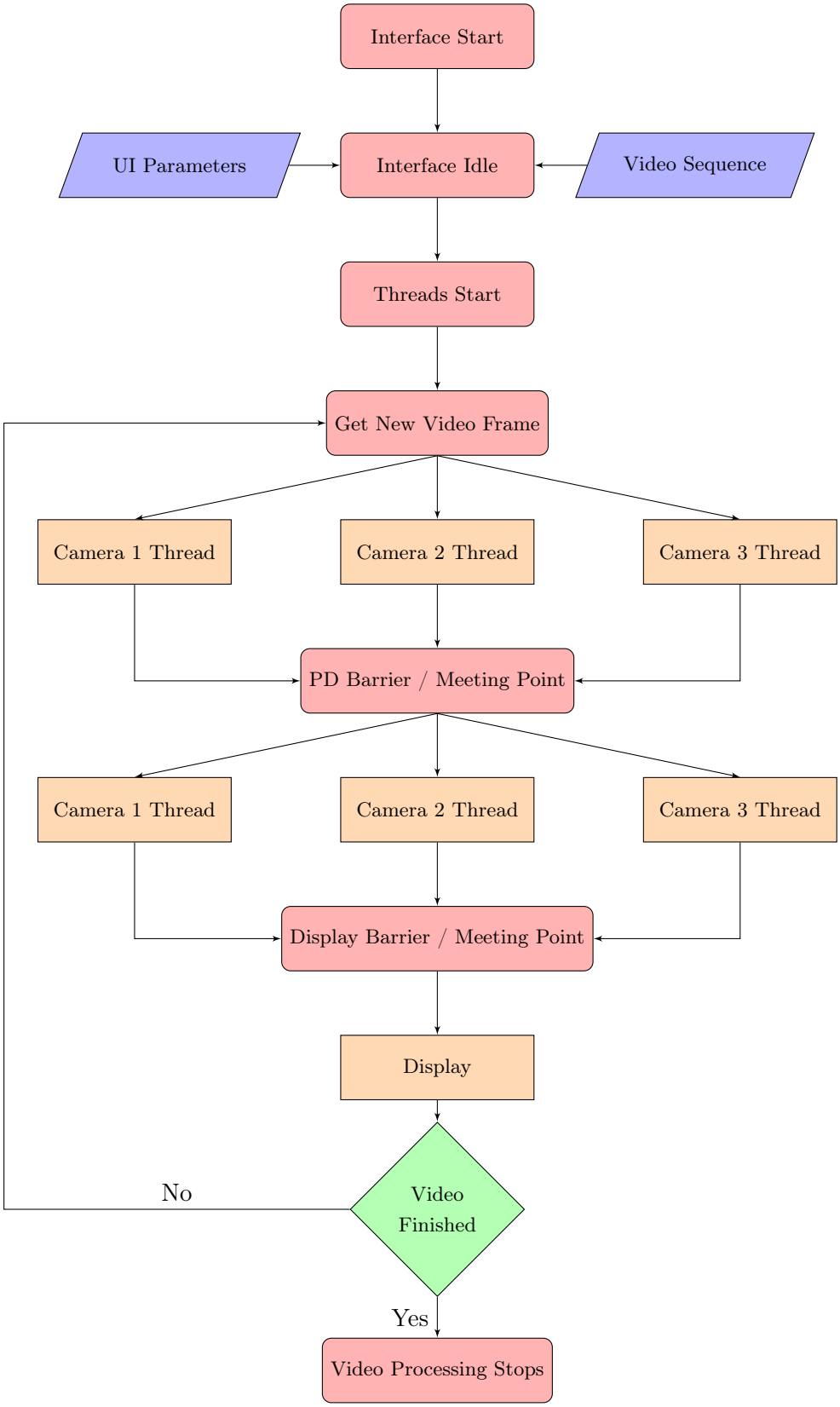


Figure 4.4: Flow-chart diagram for the multi-thread application

4.2.1 Main Application Window

Main application window is shown in Figure 4.5. As one can observe it is compose of four separate areas:

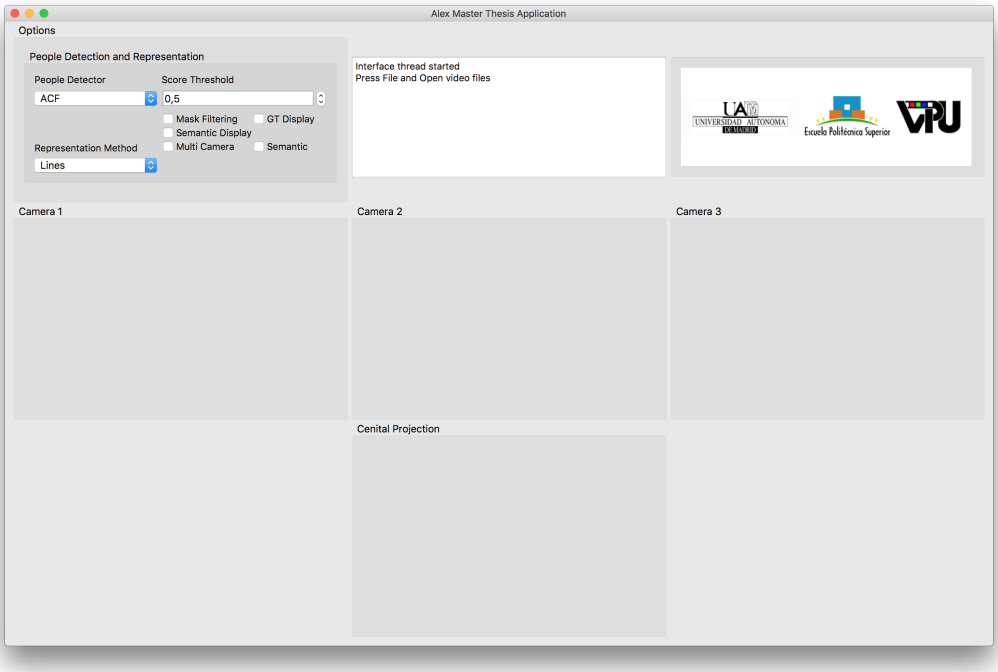


Figure 4.5: Main application window

4.2.1.1 Application Menu Bar

In the menu depicted in Figure 4.6 the main application actions are contained. From here the user can:

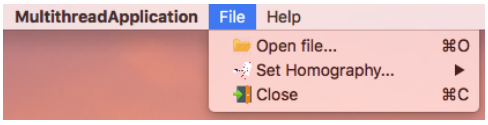
1. Open a new video sequence.

2. Compute the set of needed homographies for the integrated algorithms.

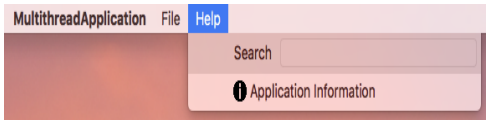
3. Close the application.

4. Search through the help searcher.

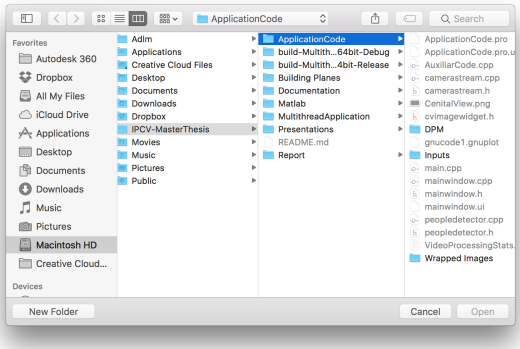
5. Open the external information window.



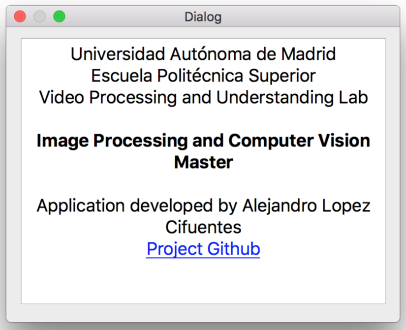
(a) File submenu



(b) Help submenu



(c) Open file dialog



(d) Application information window.

Figure 4.6: Application menu bar

4.2.1.2 Options Menu

The options box (see Figure 4.7) in the application contains all the possible parameters that can be tuned during the execution of the program. From here algorithms can be changed in real time so there is no need to restart the execution before changing some parameter. From here user can change:

- Pedestrian detectors. The user can select among the following ones:
  - ✧ PSP-Net detector
  - ✧ HOG
  - ✧ DPM
  - ✧ ACF
  - ✧ Fast-RCNN
- Different representation methods for PD detections as explained in Section 3.3.3:
  - ✧ Lines
  - ✧ Gaussians
- Enables the user to select the threshold for PD algorithms.

- PD Filtering or constraint as explained in Section 3.3.3 can also be changed. The available options are:
  - ✧ Raw PD
  - ✧ PD with semantic constraining.
  - ✧ PD with multi camera reprojection.
- Mask filtering option to perform PD over a limited area.
- Ground truth check box to display or not ground truth information.

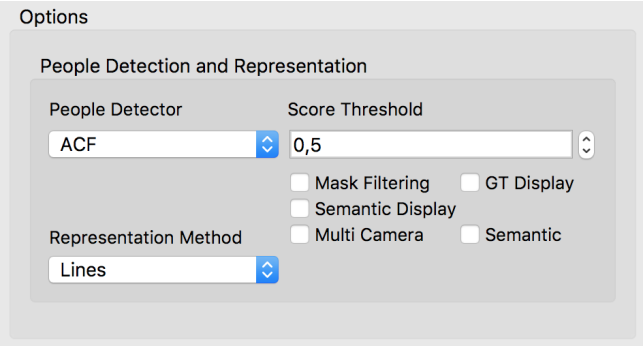


Figure 4.7: Options Menu

4.2.1.3 Information Display

Along this text box status information is provided to the user. Messages such as "Open video files", "Processing starts now" or "DPM Pedestrian Detector is now in use" appear during the execution of the application so the user can obtain some information about what to do, or what algorithm is in use. This can be observed in Figure 4.8.

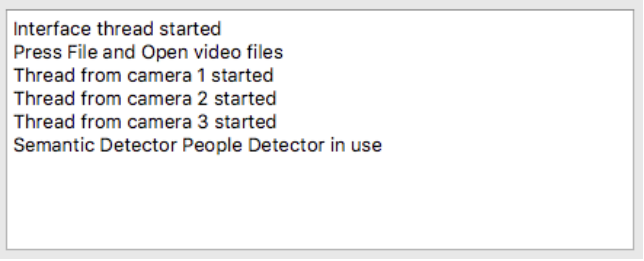


Figure 4.8: Information Display

4.2.1.4 Results Display

This is the main display area in the application in which all the visual results are presented.

We have three separate windows for each of the used cameras as well as one more display window for the cenital plane. Here the camera frames and associated detections and/or ground truth are shown: Besides all the projected semantic can be observed on the cenital frame. An example is depicted in Figure 4.9.

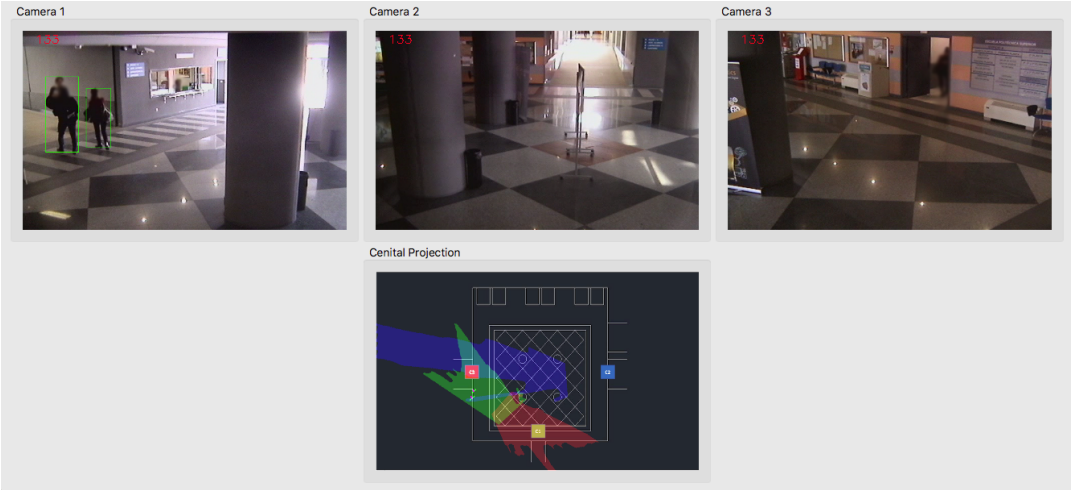


Figure 4.9: Results Display Area

4.2.2 Classes Distribution

In terms of C++ basic units the application has been divided into several classes for a better code design and to ease code comprehension / interpretation.

- **MainWindow**: This class corresponds to the main interface window and main application thread. It is the base for all the further processing as everything is inherited from this class. The reason for that is that **MainWindow** class is used to create threads and sharing procedures between them. Associated functions for this class are:
  - ✧ Creating file open dialogs.
  - ✧ Setting up and start all the camera threads.
  - ✧ Update all the algorithms configurations from the UI.
  - ✧ Displaying results through the CVImageWidget class.
  - ✧ Sharing information between threads.

- **AboutWindow**: Class that executes the second available information window. This instance displays general application information.

- **CameraWorker**: Main class for all the execution in each of the cameras. **CameraWorker** class is linked with a unique thread that process all the algorithms inside. It has **CameraStream**, **PeopleDetector**, **Evaluation** and **Barrier** classes declared within it to distribute the processing.

- **CameraStream**: This class includes all the functions that are related to video processing except PD:

  ✧ Main sequence reading loop.

  ✧ Homographies calculations

  ✧ Semantic projections.

- **PeopleDetector**: Main class to carry PD out. All the functions to detect, project and draw results either on the camera frame or on the cenital plane are in this class.

- **Evaluation**: Here ground truth is read and also the evaluation between system pedestrian detection and ground truth information is performed.

- **Barrier**: This class deals with thread synchronization. It is declared in **MainWindow** and passed by arguments to the thread so each of them has the same exact barrier object to perform the synchronization.

- **CVImageWidget**: Display representation class that deals with all the processes to draw OpenCV Mat images into the QT main window interface Widget.

In Figure 4.10 a hierarchical representation of how the different objects are arranged is presented.

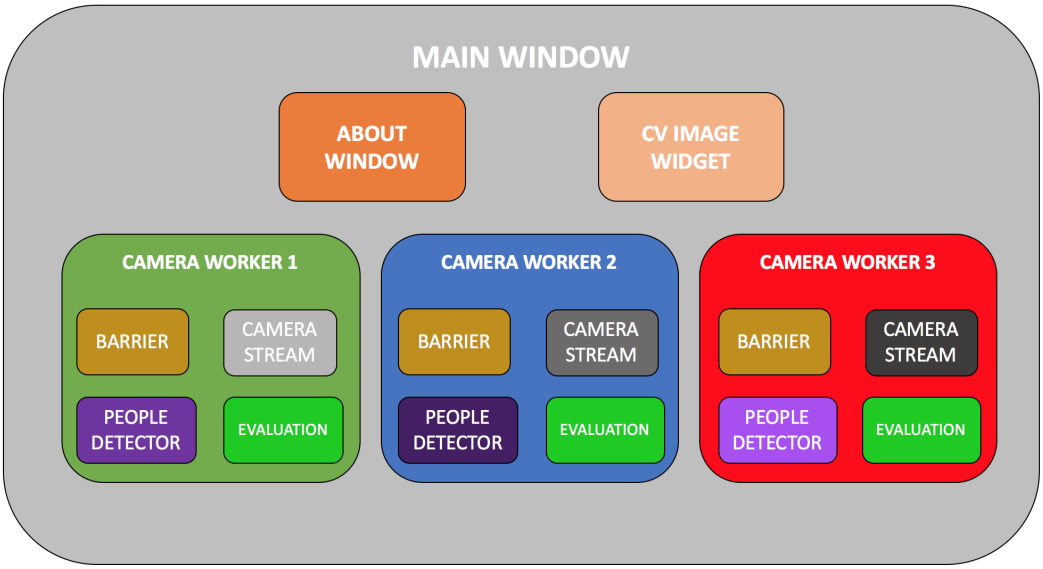


Figure 4.10: Hierarchical representation of the code

As one can observe everything is under the heritage of the ***MainWindow*** object. Here we have three ***CameraWorker*** threads that include the ***Barrier*** object, and three ***CameraStream***, ***PeopleDetector*** and ***Evaluator*** −one for each one−. In addition, ***MainWindow*** instantiates also ***AboutWindow*** and ***CVImageWidget*** objects.

4.3 Multi-thread User Application

On the contrary to the discussed version, the second developed application is focused on general users. This version only allows to load the video files and display results. All the parameters are set by default so the user does not have to fine tune any of them. This turns the usage and the general perception of the application much more simple and easy. Default setup is parametrized as the best observed configuration in the results (see Chapter 5). Figure 4.11 displays the general user application window.

Conversely to the developer application, the only functionality of the main window is to display results and guide information. All the options presented in the developer application for parameter setup are no longer available.

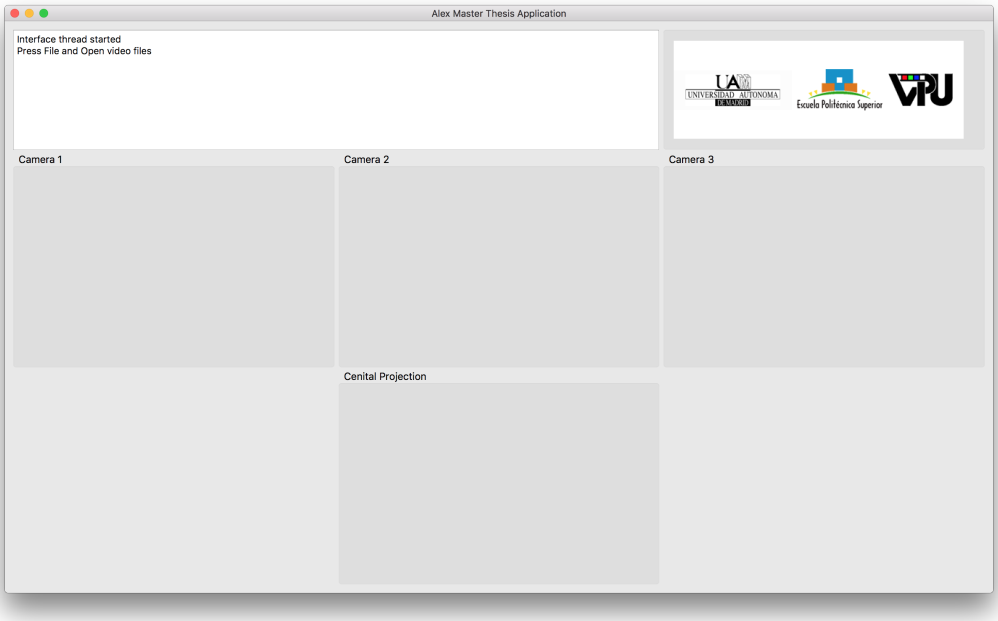


Figure 4.11: User version application main window

Chapter 5

Results

During this Chapter all the achieved results are exposed and analyzed. First, a brief analysis of the recording hardware used during the Thesis is done. Second, in detail, the experiment setup is explained, i.e. the generated data-set, ground truth, and the evaluation framework. Later, qualitative results concerning homography projections and semantic segmentation are presented. Pedestrian detection results in term of performance are also extracted. Usage data extraction, and specifically, usage curves and most used paths are analyzed. Finally, results concerning application performance are presented.

5.1 Recording Hardware

The project has been developed in the Escuela Politécnica Superior (Universidad Autónoma de Madrid). Due to this fact, the testing environment has been the hall of the mentioned engineering school which has a setup of three Internet Protocol Cameras (IP Cameras). This type of cameras can send and receive data via a computer network and Internet which allows the user to set the configuration and receive frames from the cameras.

5.1.1 Camera Specifications

Specifically, the camera model used along the project has been the Sony SNC-RZ50P PTZ Camera. This is a PTZ camera which means that is able to Pan, Tilt and Zoom all over the scene. This camera has a pan range of 340 degrees and a tilt range of 115 degrees, enabling users to monitor a wide area over the scene if the camera is moved (Figure 5.1) .

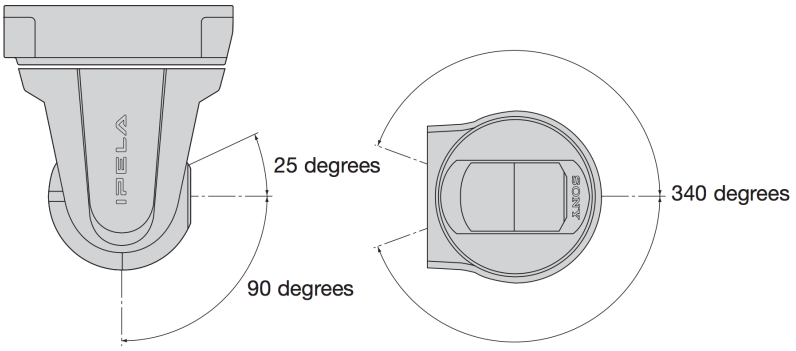


Figure 5.1: Camera Sony SNC-RZ50P Pan/Tilt Range diagram

The complete and relevant specifications of the cameras are detailed in Table 5.1. In the scope of our work the most important features are the frame resolution and the number of frames per second. The highest resolution and highest frame rate profiles have been chosen to operate on the best quality available scenario.

Camera	
Horizontal viewing angle	1.7 to 42.0 degrees
Focal length	f = 3.5 to 91.0 mm
F-number	F1.6 (wide), F3.8 (tele)
Minimum object distance	320 mm (wide),1,500 mm (tele)
Pan angle	-170 to +170 degrees
Pan speed	300 degrees/s (max.)
Tilt angle	-90 to +25 degrees
Tilt speed	300 degrees/s (max.)

Image		
Image size (H x V)	**640 x 480, 320 x 240, 160 x 120**	
Compression format	JPEG, MPEG-4, H.264	
Maximum frame rate	**JPEG/MPEG-4**	**25 fps (640 x 480)**
	H.264	**8 fps (640 x 480)**

Table 5.1: Camera Sony SNC-RZ50P Specifications

5.1.2 Camera User Web Interface

The camera comes with a built-in web interface that helps the user to visualize the visual range and set the different parameters that would change the camera behavior. The most important features that users are able to tune are described as follow:

- Camera control: Through this interface one can control and set the position of the camera in terms of pan, tilt and zoom (Figure 5.2). Changes in this three variables lead to different visualizations of the scene.

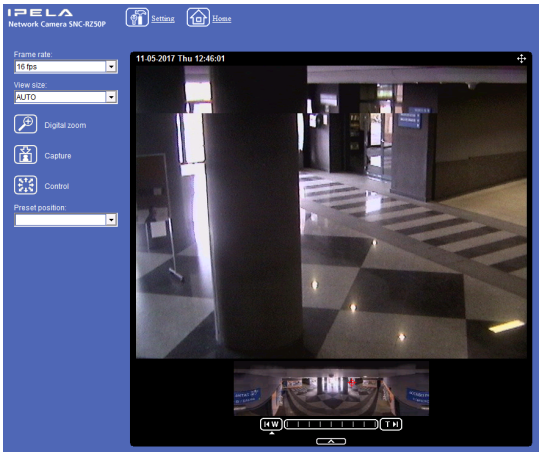


Figure 5.2: Visualization and control menu

- Preset position: In this menu (Figure 5.3) one could save the position that has been set in Figure 5.2 in order to recover the same position if the camera has been moved before in precise and easily manner.

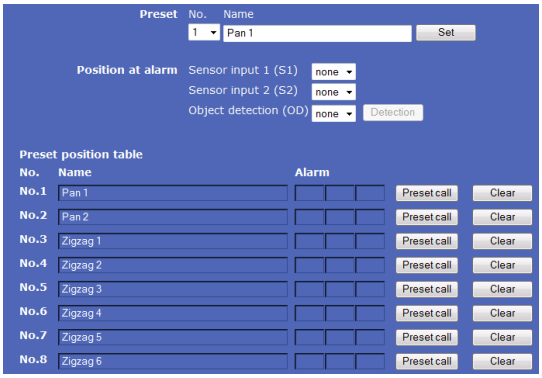


Figure 5.3: Preset position setting menu

- Tour setting: One can set the camera to describe a tour over the scene. This process is done by setting at least two different preset positions from where the

camera is moving from one to the other at a also configurable set speed. The menu to configure this behavior is displayed in Figure 5.4.

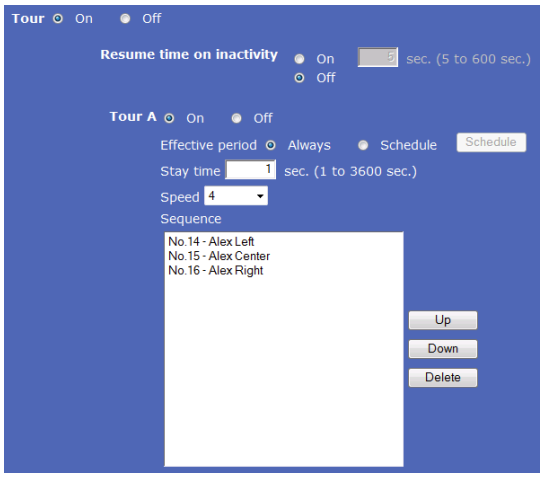


Figure 5.4: Tour setting menu

5.2 Experimental Setup

In this section, the generated data-set, ground truth and the proposed evaluation framework used to obtain results are detailed. All these factors apply to all the performed experiments.

5.2.1 Data-Set

Data-set has been generated in the university hall. A tour for each camera has been set so they move only changing position in the X/horizontal axis and not in the Y/vertical direction. Reasons concerning camera panning were presented in Section 3.3.1.3.

The complete data-set consists on a set of three different scenarios of $5-8$ minutes long recorded on three different days. In each of the scenarios sequences for all the cameras have been obtained −i.e. 3 videos for each sequence, one for each camera (see Figure 5.5 for an example)−. The technical characteristics of the videos are:

- Number of videos per recording: 3
- Resolution: 640x480 pixels.
- Frame rate: 23.976 fps.
- Video Format: MPEG Video (Version 2) (Main@High).

Figure 5.5: Data-set example frames for a recorded scenario

5.2.2 Ground Truth Generation

We have selected one of the three scenarios to fully evaluate the proposed system.

The selected sequence for evaluation purposes is a mixture between easy and complex situations for a pedestrian detector. Situations range from people walking alone across the hall, to people pushing objects like a wheelchair. The scenario also includes big groups of people both inside and outside the building. In general due to image quality in terms of resolution and illumination and pedestrian situations the selected sequence is in the medium-high range of complexity for pedestrian detection (see Figure 5.6 for examples frames of the selected scenario).

In order to carry the evaluation out ground truth information is needed. The video has been manually annotated with Via Annotation Tool[1] to generate pedestrian bounding boxes. This software allows the user to generate XML files with bounding boxes positions for complete sequences. The scenario is composed of three different videos of 8300 frames, which means that the total number of manually annotated frames is 25.200.

[1] https://sourceforge.net/projects/via-tool/

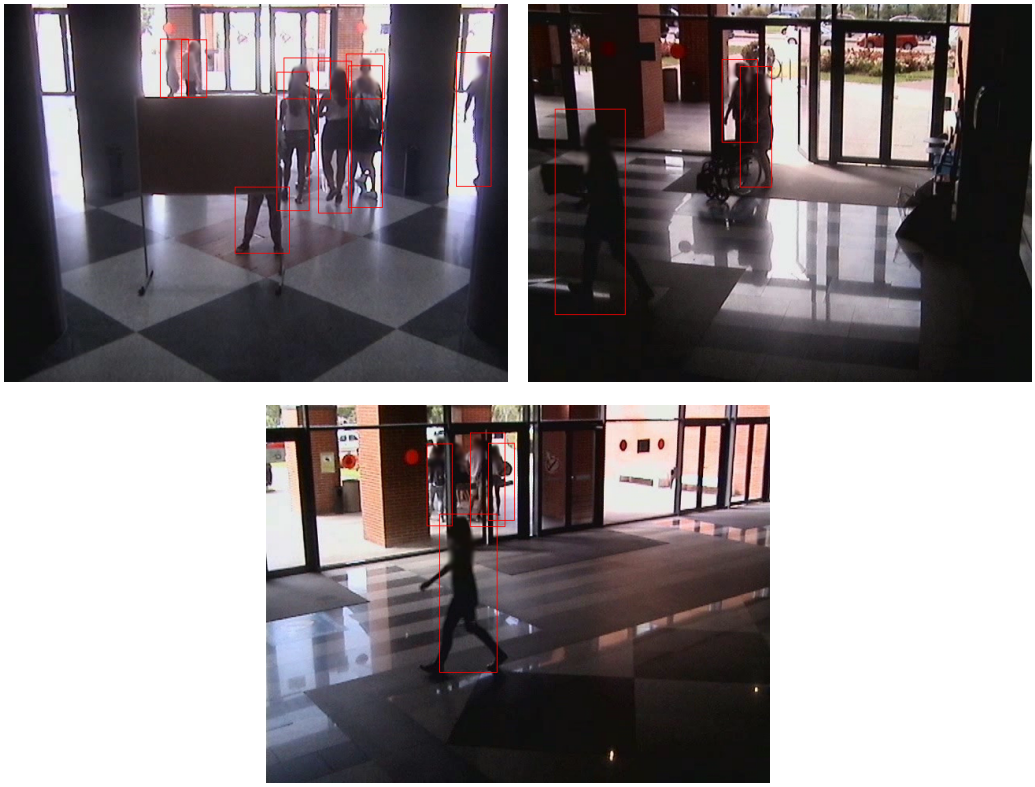


Figure 5.6: Annotated ground truth frames

5.2.3 Evaluation Framework

Evaluation Metrics

Once the data-set scenario has been correctly annotated one can proceed to evaluate the performance of pedestrian detectors. In order to evaluate these algorithms Precision and Recall metrics are used. The calculation of these metrics is defined in Eq. 5.1 and Eq. 5.2, respectively.

$$Precision = \frac{\#True\,Positives}{\#True\,Positives + \#False\,Positives} \tag{5.1}$$

$$Recall = \frac{\#True\,Positives}{\#True\,Positives + \#False\,Negatives} \tag{5.2}$$

In addition, to measure performance in a more general process F-Score is used. It combines both Precision and Recall in a unique metric measure. Eq 5.3 express the general metric expression.

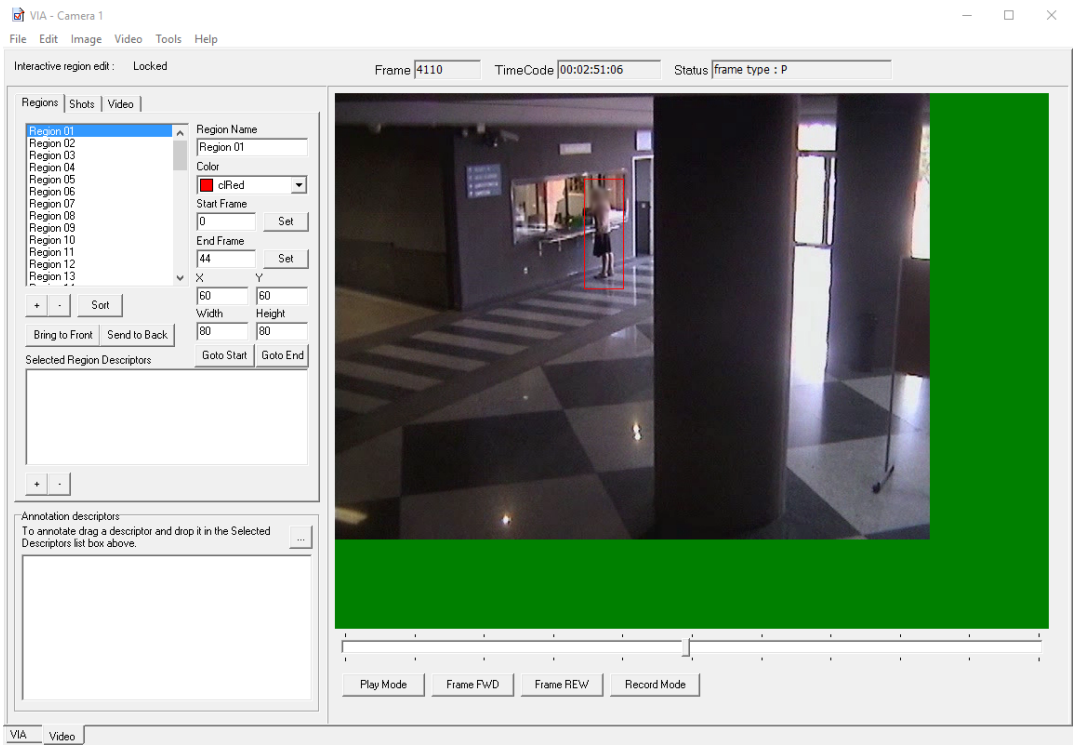


Figure 5.7: Via Annotation Tool software main window

$$F - Score = \frac{(2 \cdot Precision \cdot Recall)}{(Precision + Recall)} \tag{5.3}$$

Bounding-Box Comparison

To extract these metrics we should have a method to compute which bounding boxes match with the ground truth. In order to achieve this objective Intersection over Union (IoU) or Jaccard metric has been selected. IoU is computed as in Eq 5.4 and it measures the relation between bounding boxes overlapping and its union areas.

$$IoU(Detection, GroundTruth) = \frac{OverlapArea(Detection, GroundTruth)}{UnionArea(Detection, GroundTruth)} \tag{5.4}$$

This means that even if the bounding boxes overlap perfectly −one lies completely inside the other−, but the union area of them is high, IoU metric has a small value. This effect can be observed in Figure 5.8.

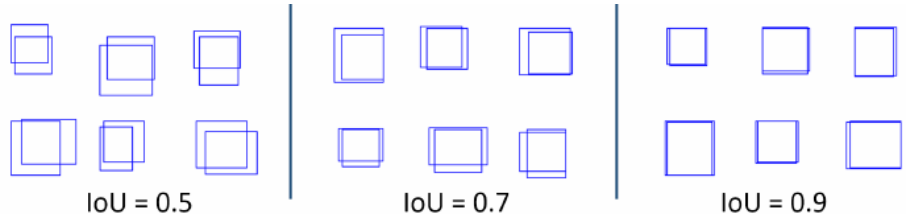


Figure 5.8: Intersection over Union or Jaccard Index

For our evaluation framework a detection is consider correct if

$$IoU(Detection, GroundTruth) \leq 0.5$$

which is the usually selected metric value in the State of the Art.

5.3 Homography and Semantic Segmentation

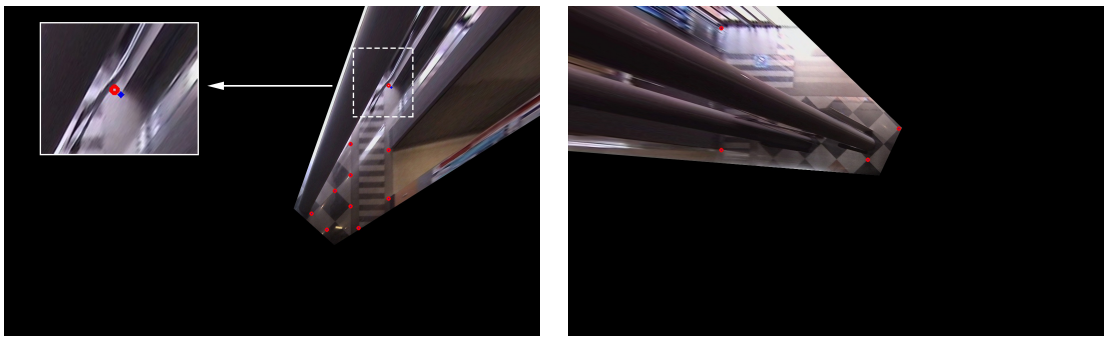
During this Section qualitative results regarding homography and semantic segmentation calculation are presented.

Homography

Correctly calculated homographies are essential for the right performance of further algorithms. In order to test that homographies have been correctly achieved by the manual selection of points, views have been projected with its correspondent matrix. In addition, selected points from both the frame (blue points) and the cenital plane (red points) have been overlapped in the same image. Some examples can be observed in Figure 5.9.

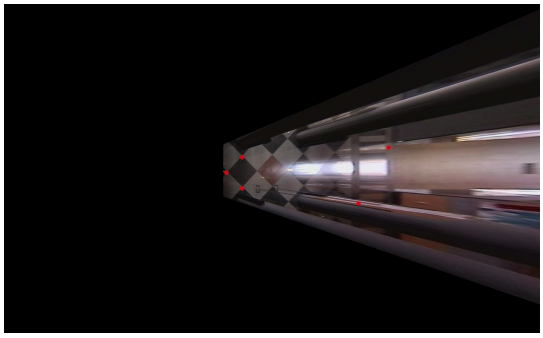
Qualitative results in Figure 5.9a suggest that homographies are accurate enough in almost all the cases. Blue points are not visible because of the overlapping with the red ones, which means that their correspondence is perfect. However, one small error can be noticed in Figure 5.9a where there exists a displacement between the red point and the blue one (top of the image).

It is important to highlight, how precision and image quality is reduced the further the points are from the camera. It can be observed in the blurred image parts of the images. This is a standard problem when dealing with such type of homographies.



(a) Camera 1 View 7

(b) Camera 2 View 6



(c) Camera 3 View 5

Figure 5.9: Projected views with overlapping of selected points.

Semantic Segmentation

In this Section results regarding the performance of PSP-Net in the proposed scenario are presented. In order to test the algorithm and for the lack of ground-truth annotations for semantic segmentation, we include in Figure 5.10 illustrative examples of common problems that should be qualitatively evaluated.

Taking a look to results one can derive that PSP-Net extracts high quality results in terms of semantic. It is noticeable that most of the floor, if it is not occluded by some scene element, as in the top image from Figure 5.10, is correctly extracted.

It is also important to point out that segmentation is not perfect when dealing with difficult illumination cases or complex objects. It can be observed that columns are not correctly segmented. Some of them are misclassified as walls or floor. This is one of the main issues for our system as an accurate floor detection is needed for constraining pedestrian detections. In addition, crystal doors such as the ones displayed in the top and bottom images, are not correctly distinguish from building and wall labels which could also be a main issue for further statistical data extraction algorithms.

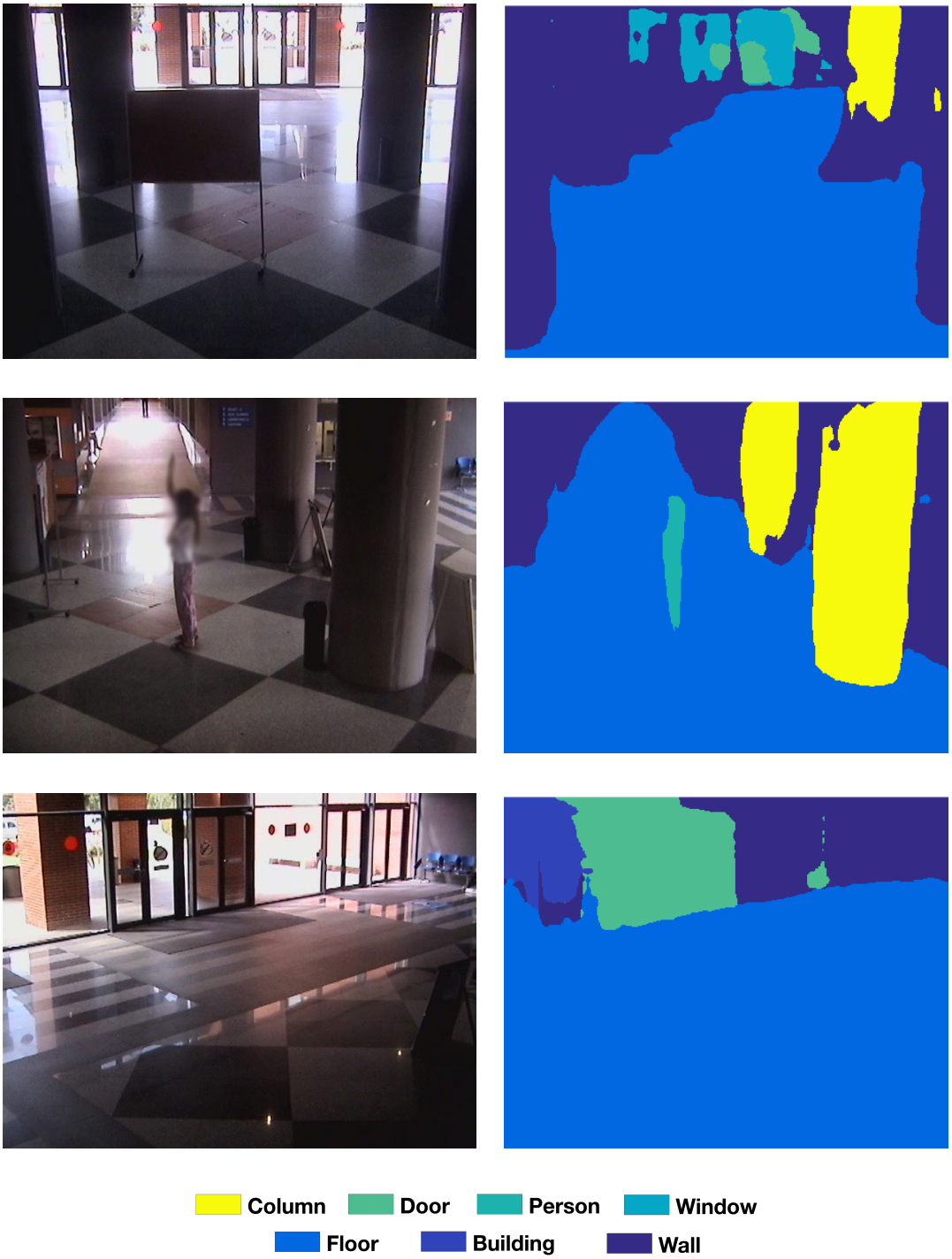


Figure 5.10: RGB frames and PSP-Net Semantic segmentation results. From top to bottom: Camera 1, Camera 2, Camera 3.

5.4 Pedestrian Detection

During this section results concerning PD approaches are presented. HOG, DPM and PSP-Net algorithms are chosen for evaluation. However, the ACF algorithm is also integrated in the application and offline generated results from Fast-RCNN can be also load in the application.

We propose a set of four different experiments to measure the overall performance of the developed system and the contribution to PD of each of the designed modules.

Results are presented within Recall versus Precision curves. In order to create these graphs, algorithms have been evaluated in all the four test using five different values from the score thresholding $[0, 1]$ interval. Each threshold iteration (0, 0.2, 0.4, 0.6, 0.8) for an algorithm in a test, provides one point of the curve. To proceed with this process all the algorithms scores have been normalized from its original values to the $[0, 1]$ interval. Maximum and minimum scores have been used for this purpose. In order to obtain scores extrema for all the detectors a train sequence selected from the data-set has been used.

Mono-Camera Environment

Here pedestrian detections are tested working only in a mono-camera environment. This means that neither information is shared by the cameras nor semantic constrains are applied to detections. This experiment measures performance of raw detectors. Results are presented in Figure 5.11.

Multi-Camera Environment

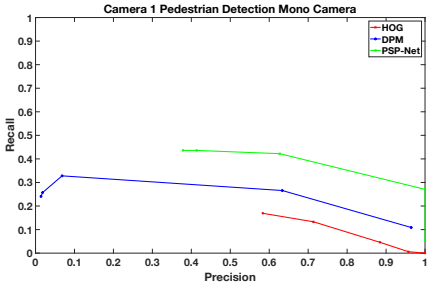
Algorithms are tested in a multi-camera setup. As explained in 3.3.3.2 detections from cameras are reprojected and combined onto other cameras . Results are presented in Figure 5.12.

Mono-Camera Environment with Semantic Constraining

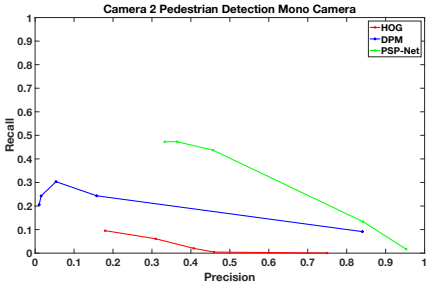
People detection are analyzed again in a mono camera environment but this time, applying semantic constraining as explained in Section 3.4. Results are presented in Figure 5.13.

Multi-Camera Environment with Semantic Constraining

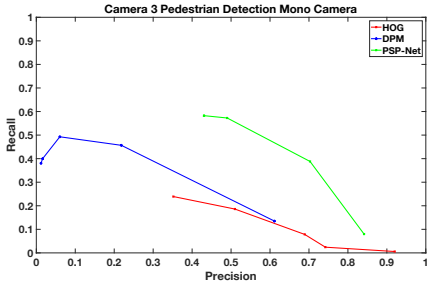
Finally, the last test embraces all the proposed algorithms running within the multi-camera reprojection setup and applying semantic constraints. (Figure 5.14)



(a) Camera 1

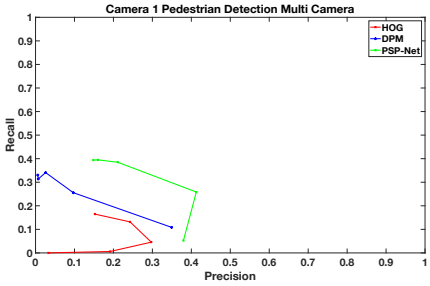


(b) Camera 2

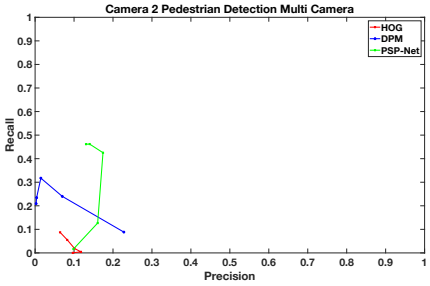


(c) Camera 3

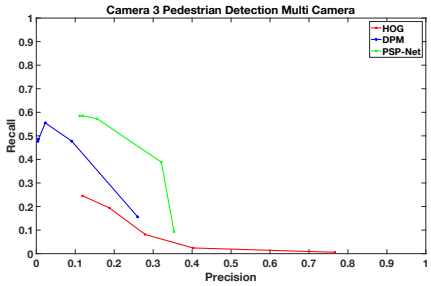
Figure 5.11: Recall / Precision graphs for mono camera pedestrian detection



(a) Camera 1

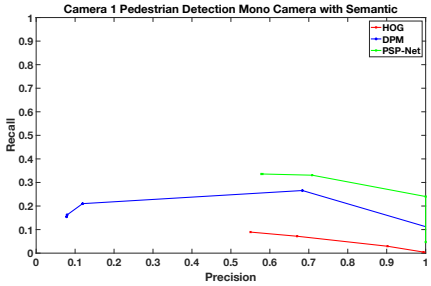


(b) Camera 2

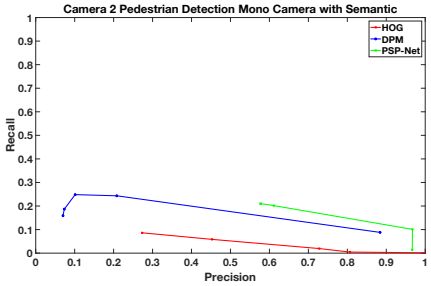


(c) Camera 3

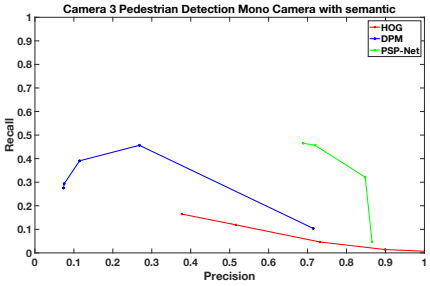
Figure 5.12: Recall / Precision graphs for multi camera pedestrian detection



(a) Camera 1

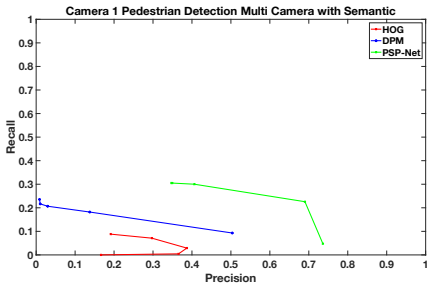


(b) Camera 2

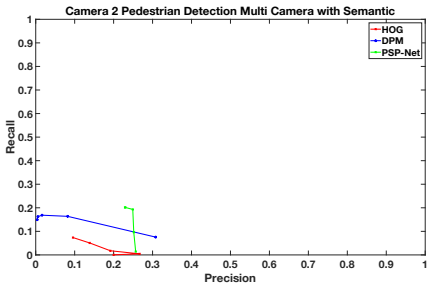


(c) Camera 3

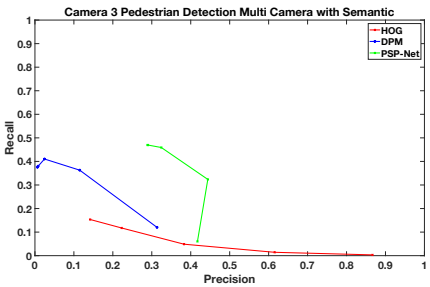
Figure 5.13: Recall / Precision graphs for mono camera with semantic constraining



(a) Camera 1



(b) Camera 2



(c) Camera 3

Figure 5.14: Recall / Precision graphs for multi camera with semantic constraining

Overall Comparison

In order to compare numerically all the PD approaches in different experiments F-Score results have been obtained and presented in Table 5.2. For this Table the best F-Score metric for a PD for each particular experiment has been chosen.

Pedestrian Detector		Mono Camera	Multi Camera	Mono Camera Semantic	Multi Camera Semantic
HOG	Camera 1	**0.26**	0.17	0.15	0.12
	Camera 2	0.12	0.07	**0.14**	0.08
	Camera 3	**0.28**	0.19	0.22	0.15
DPM	Camera 1	0.37	0.16	**0.39**	0.16
	Camera 2	0.19	0.12	**0.23**	0.13
	Camera 3	0.29	0.19	**0.34**	0.17
PSP-Net	Camera 1	**0.50**	0.31	0.45	0.34
	Camera 2	**0.44**	0.24	0.30	0.22
	Camera 3	0.52	0.35	**0.56**	0.38

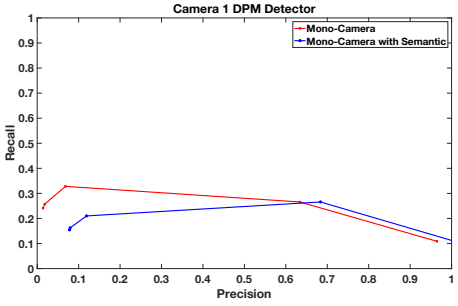
Table 5.2: F-Score comparison between all the experiments and pedestrian detector approaches. Best results for an approach in each camera is in bold. Best overall result is in red.

5.4.1 Pedestrian Detection Results Discussion

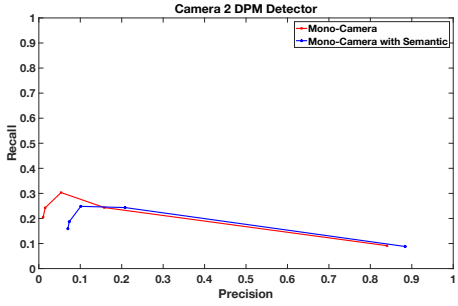
General PD Performance

If one analyzes Figures 5.11, 5.12, 5.13 and 5.14 it is easily observable that PSP-Net outperforms DPM and HOG in every experiment and in all the cameras. This is quite logical as algorithms based on CNNs usually perform better than classical PD approaches (see Chapter 2).

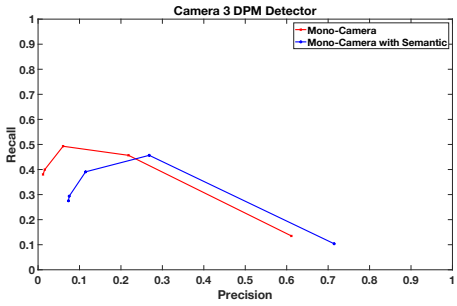
However, none of the detectors yields adequate results for the test sequence. Ideal results should lead to high recall and low precision values when $Th \simeq 0$ and on the



(a) Camera 1



(b) Camera 2



(c) Camera 3

Figure 5.15: Recall / Precision curves for DPM in different experiments

other hand, low recall and high precision values when $Th \simeq 1$. This poor results may be influenced by the sequence difficulty and the complete ground truth.

Effect of Semantic Constraining

If one focus the analysis on the general system performance, and specifically, considering the effect of semantic constraining we have to center the attention to Table 5.2.

If we take a look at HOG descriptor two out of the three best results are obtained using raw HOG. Only Camera 2 with semantic constrain marginally outperforms F-Score of the raw detector. PSP-Net presents the same effect where only camera 3 gets a better result using semantic constraining. However, DPM using semantic filtering outperforms the scores obtained with raw DPM for every camera, suggesting that the inclusion of semantic information leads to and increase in DPM performance. This effect can also be observed when Precision and Recall values for DPM detector in two different experiments are depicted in the same Recall/Precision graph (Figure 5.15).

In our opinion, HOG and PSP-Net do not appear to benefit from semantic segmentation due to their lower number of detections with respect to DPM. Both approaches,

(a) DPM (b) HOG

(c) PSP-Net

Figure 5.16: Differences in number of detections for $Th = 0$ and same t.

although their scores have been normalized to the [0, 1] interval, extract good results even with the lowest threshold. This is totally contrary to DPM detector, in which, using $Th = 0$ a high number of detections are obtained. (See Figure 5.16 for visual results).

This means, that if detections are already accurate and due to our common semantic maps, they lay on some of the floor holes from Figure 3.21, detections are suppressed. This leads to obtaining a worse performance than using the raw detector. In order words, semantic constraining is to be consider worthwhile if the number of false positives laying in areas different than floor is higher than the number of wrongly suppressed detections.

In addition, as depicted in Figure 5.6 ground truth has been made so every pedestrian is annotated in the sequence. This is a problem when using semantic constraining as every detection placed at the outside is suppressed for not being inside the hall. Furthermore, PSP-Net as is depicted in Figure 5.17 does not obtain pedestrian detections when analyzing people outside the building which could be one of the reasons of its low performance.

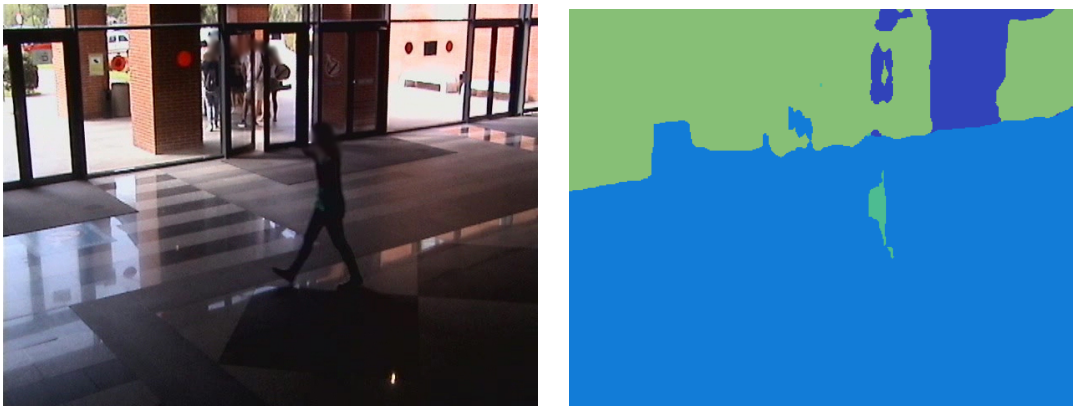
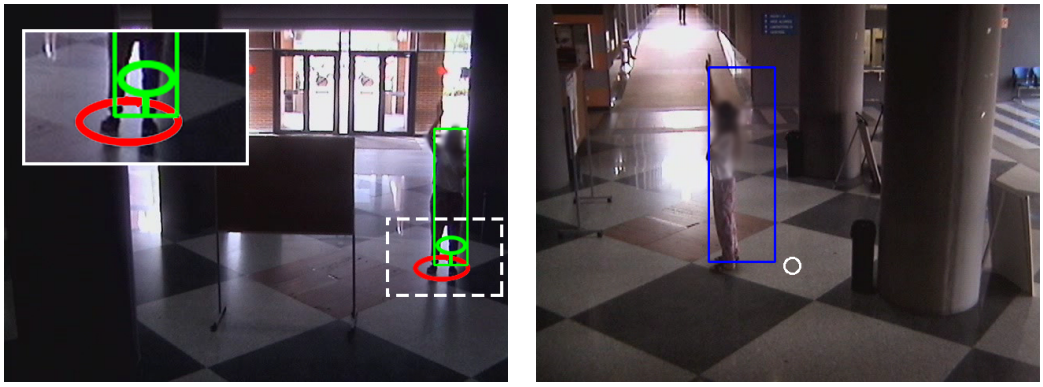


Figure 5.17: Misclassification of people in semantic segmentation. Camera 3



(a) Camera 1 (b) Camera 2

Figure 5.18: Pedestrian detection error leads to a reprojection displacement. Camera 1: Red circle corresponds to correct cylinder and green one corresponds to the actual estimation. Camera 2: White circle represents reprojection from Camera 1.

Multi Camera Reprojection System Performance

If we now focus on the performance of the multi camera system when dealing with reprojections of pedestrian, Table 5.2 shows that is not properly working. In both experiments in which reprojection is used (second and fourth experiments) worse results than even the raw detectors are obtained. This could be due to two reasons:

1. If a person is detected but, the bounding box does not surround entirely the person, the reprojection is not accurate. This effect can be observed in Figure 5.18, in which the detection is correct but feet are not in the bounding box. This leads to a spatial difference between the cylinder center estimation and the actual person (see Figure 5.18b).

(a) Camera 2 (b) Camera 3

Figure 5.19: Pedestrian reprojection error due to height

2. When reprojecting detections, blob height is lost and further reconstructed with respect to a fixed aspect ratio. This process could lead to examples as the ones in Figure 5.19. Projections are finally reconstructed as small or big blobs if width is small or big respectively. Two reprojections from other cameras are present however, one of them has not been combined. This means that reprojections not correctly combined with the original detection from the camera produce a false positive.

5.5 Statistical Usage Data

In this Section we present a couple of simple applications that exploits the extraction of statistical usage data.

5.5.1 Experiment 1

We propose to illustrate usage curves on two different semantic areas: *doors* and *floor*. The choice has been made regarding the limitations of the map in Figure 3.21. On each semantic area one pedestrian density per frame graphic is obtained. The obtained curves are displayed in Figures 5.20 and 5.21.

5.5.2 Experiment 1 Results Discussion

Observing Figure 5.20 and considering the correspondent frame, displayed at the top of the graph, it can be observed that there is a correspondence between periods when most people are walking through the scene and the different peaks from the curve. This means that promising results in this area are obtained. Nevertheless, Figure

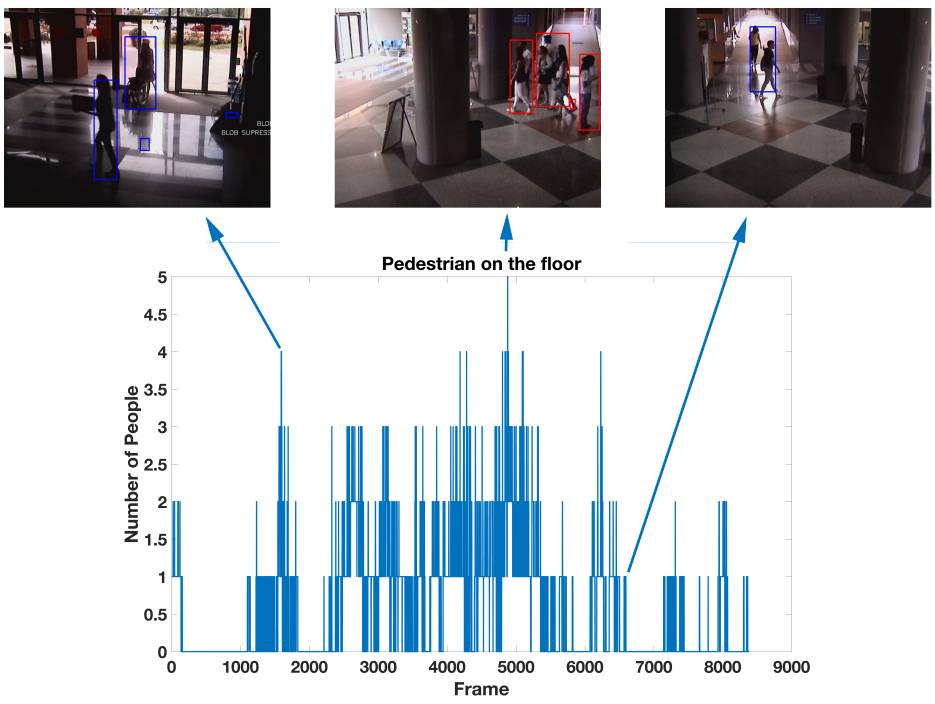


Figure 5.20: Floor usage graph. From left to right in camera frames: $t_1 = 1594$, $t_2 = 4877$ and $t_3 = 6228$.

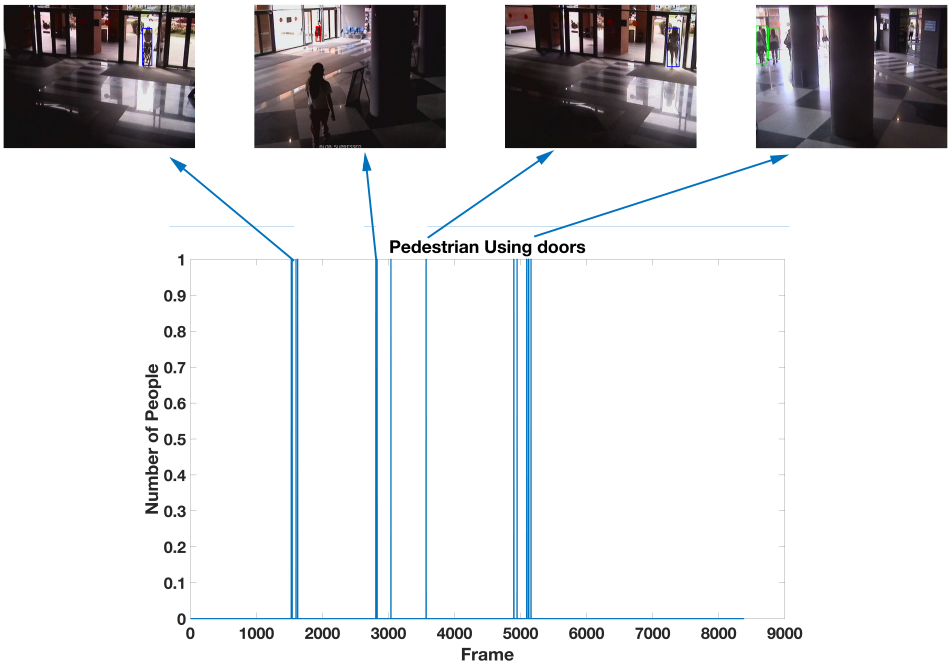


Figure 5.21: Doors usage graph. From left to right in camera frames: $t_1 = 1541$, $t_2 = 2808$, $t_3 = 3569$ and $t_4 = 5091$.

5.20 at $t_1 = 1594$ and $t_2 = 4877$ shows that we are not getting an accurate measure at t_1 and t_2 due to false positive detections, and also due to the group detection as a whole.

Regarding the *doors* usage one can observe that there an accurate relation between people passing trough *doors* in approximately frames $t_1 = 1541$, $t_2 = 2808$, $t_3 = 3569$ and $t_4 = 5091$ and curve peaks in Figure 5.21.

5.5.3 Experiment 2

In addition, some results about the most used areas of the hall have been extracted. All the scene has been divided into regular sub regions of fixed size in which pedestrian flow has been measured. Results concerning this process are displayed in Figure 5.22 for a subregion size of $40x40$ pixels and in Figure 5.23 for a subregion size of $70x70$ pixels.

Yellow squares represent the amount of pedestrians found in that sub region. The more strong the color appears in the image it means that pedestrian density over that area has been higher than in the neighborhood. A temporal comparison has been made with the correspondent frame.

5.5.4 Experiment 2 Results Discussion

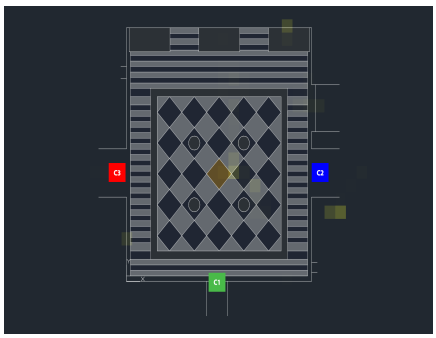
In both Figures 5.22 and 5.23 it is observable that most used paths increase accordingly pedestrian flow in the sequence. For that reason when a person enters the building areas surrounding the door get highlighted. In the same manner, when a big pedestrian group walks along the scene all their trajectory is being highlighted as pedestrian density over that areas is highly increased.

5.6 Application Performance

Along this section the experiments concerning the application efficiency are described.

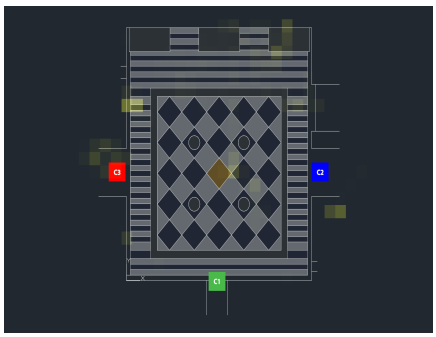
Firstly, results concerning differences between the single thread and the multi thread applications are exposed. Measures have been obtained by using the same sequence and the same PD algorithms. In Table 5.3 one can observe the difference in terms frame per second for the same video sequence and the same pedestrian detector approach (HOG).

(a) Camera 2 . Frame at $t = t_0$



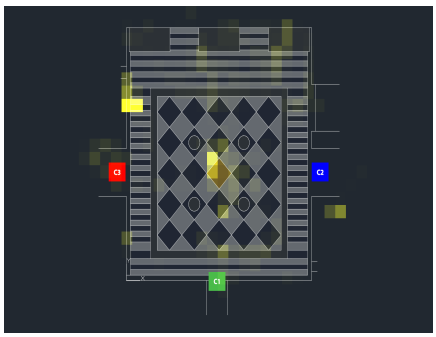
(b) Usage Map at $t = t_0$

(c) Camera 1. Frame at $t = t_1$



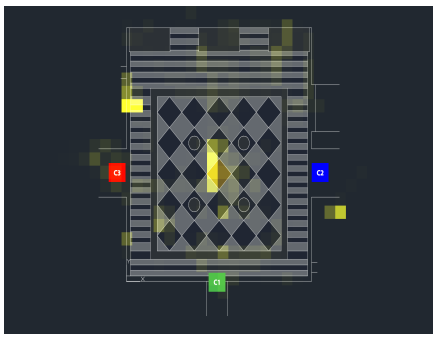
(d) Usage Map at $t = t_1$

(e) Camera 1. Frame at $t = t_2$



(f) Usage Map at $t = t_2$

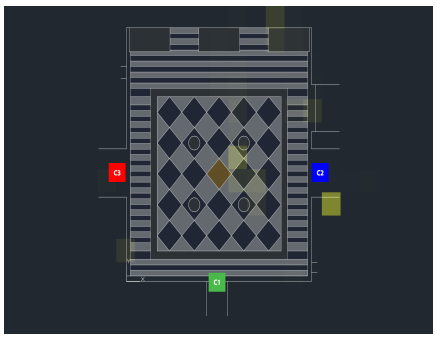
(g) Camera 3. Frame at $t = t_3$



(h) Usage Map at $t = t_3$

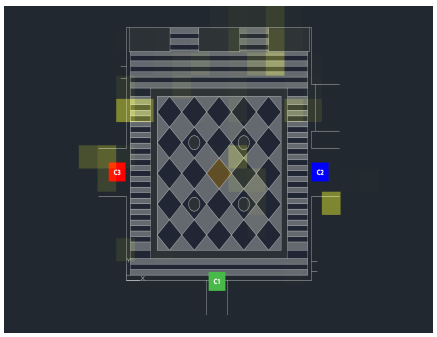
Figure 5.22: Pedestrians paths usage along the Hall. Scene has been divided in sub regions of 40 pixels.

(a) Camera 2. Frame at $t = t_0$



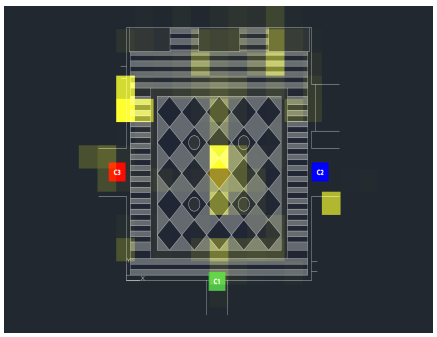
(b) Usage Map at $t = t_0$

(c) Camera 1. Frame at $t = t_1$



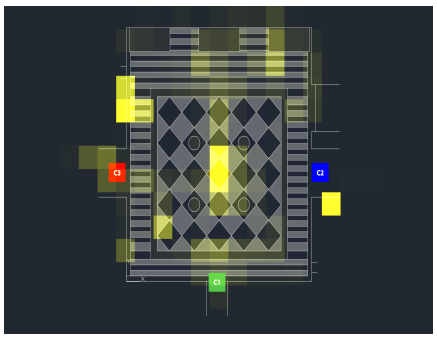
(d) Usage Map at $t = t_1$

(e) Camera 1. Frame at $t = t_2$



(f) Usage Map at $t = t_2$

(g) Camera 3. Frame at $t = t_3$



(h) Usage Map at $t = t_3$

Figure 5.23: Pedestrians paths usage along the Hall. Scene has been divided in sub regions of 70 pixels.

Programming Approach	FPS
Single-Thread	0.26 fps
Multi-Thread	**1.42 fps**

Table 5.3: Comparison between the use of single thread or multi threads

In addition, all the PD are compared when evaluated in the multi thread environment with the same video sequence. This results can be seen in Table 5.4

Pedestrian Detector	FPS
HOG	1.35 fps
ACF	0.60 fps
DPM	0.94 fps
PSP-Net (Offline)	**1.60 fps**

Table 5.4: Comparison between the use of single thread or multi threads

5.6.1 Application Performance Results Discussion

Looking at Table 5.3, one can easily derive that the speed up of using multithreads is clear. This difference implies that in the amount of time the single thread application obtains results for three frames, the multi-thread application has been executed almost 5.5 times. This is a huge improvement in the computational time.

In terms of pedestrian detection comparison, taking a look a the table one can easily observe that PSP-Net is the one which runs faster. However, its offline computation makes it evidentially the fastest. Comparing those which are truly computed online, HOG is the one that performs faster followed by DPM and ACF.

Nevertheless, it is noticeable that even the slowest PD multi thread approach, performs faster than HOG in the single-thread application.

5.7 Overall Discussion

Taking into account all the presented results one can observe that some proposed algorithms from the system are performing correctly while others need to be improved.

Homography calculation is accurately enough for the purpose of the system. We observed that almost all the homographies were correctly computed and thee only one small error does not lead to further problems in algorithms.

Semantic segmentation using PSP-Net has some main drawbacks when dealing with columns or doors segmentation. In the first case, when misclassifying columns, sometimes, leads to a false floor label. This means that finally, when creating the semantic reference plane π_{ref} some floor areas truly columns. Doors errors in some specific frames could lead to not obtaining door areas, however, when combining all the sequence frames by the temporal median filter this problem is solved as depicted in Figure 3.21.

Moving to performance in terms of pedestrian detection above results show that multi cameras reprojection is not working properly in any of the tested approaches. However, semantic constraining performs promising when dealing with algorithms that need some extra filtering due to the high amount of extracted blobs.

Semantic usage extraction, in terms of both usage curves and density paths, performs good although errors from PD are trespassed to these results and so, they are influence by PD.

Finally, application performance results completely supports the multithread application developing due to the high speedup.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This master thesis has described a system capable of performing pedestrian detection and semantic segmentation over a multi camera setup at the Escuela Politécnica Superior, Universidad Autonoma de Madrid (Spain). Different pedestrian detection approaches such as HOG, DPM, ACF or Fast-RCNN and a semantic segmentation algorithm, PSP-Net, have been selected among others from a complete study of the state of the art.

A complete system to perform pedestrian fusion and filtering has been proposed. Information from three cameras is combined into a common cenital plane by the use of homographies. This process allows PD to be shared and reprojected from one camera to another. In addition, common multi camera semantic information has been used to constraint PD detections.

Statistical data usage from different semantic areas has been extracted. Also, pedestrian paths in terms of density have been computed based on the scene division in a regular grid.

In order to control and tune algorithms a multi thread application has been developed under QT Developing environment. It has a Graphical User Interface that sets the base for a user-friendly interaction between the user and the software. This application also allows the user to represent and visualize all the extracted results.

All the system has been tested in a manually annotated recording obtaining performance results for homography calculation, PD approaches under different environments and semantic areas statistical usage data. During these tests semantic constraining has improved DPM approach, however HOG or PSP-Net algorithms constraint with semantic maps have obtained worse results than the raw detector.

Similarly, multi-camera fusion has not been able to increase performance for any of the algorithms. Problems related to pedestrian reprojection between cameras combined with PD errors lead to an increase in the number of false positive and so, in performance decrease.

Statistical data usage has been almost perfectly extracted, having exact frames in which semantic areas are used. In addition, semantic density paths have been obtained with high precision in spatial and temporally terms.

6.2 Future Work

Considering current state of the art, obtained results and extracted conclusions one can set the stage for future work.

In terms of application and software development some improvements are proposed to be done. Nowadays, heavy computational work is achieved almost in real time by the use of graphical cards. GPU computation may be implemented in the scope of this work. Many of the used methods are also implemented with GPU computation functionalities and the inclusion of this kind of speed-up may result in a improvement in the system efficiency.

Furthermore, the view selection scheme described in Section 3.3.1.5 has a substantial impact on the method's performance. However, when high illumination changes occur this process fails. We propose to fix this problem by the use of camera spatial positions. With this information exact same position for each of the views can be obtained periodically and so, they can be updated during the video sequence in order to adapt to illumination changes.

One of the main problems discussed in the results section is that when pedestrians are reprojected the blob height is lost. Due to this, there are some frames that have small detections compared to the person size. To correctly reproject the blob we propose as future work to use real distances between camera and pedestrian detection to finally obtain an approximation to the real height for the blob.

Regarding results, other data-sets scenarios may be evaluated and ACF and Fast-RCNN detectors may need to be included in the tests.

Finally, we propose to use the Gaussian representation of bounding boxes to perform pedestrian fusion over the cenital plane instead of performing it on the camera frames.

Appendix A

Cenital Plane Design

One of the main objectives of the work is to project extracted detections from the three cameras, i.e. pedestrian and semantic, into a common plane for all of them. To achieve this goal a cenital plane that correctly represents the scene is needed.

First Approach

The first used approach to represent the cenital plane is depict in Figure A.1.

Figure A.1: First cenital plane approach

This cenital plane lacks of details from the scene. The information about the details of the scenario is minimum and also, the scene proportions are not correct. To

compute a correct homography between the camera frame and the cenital plane one should be able to identify the same scene points in both images in the ground plane. This means that the cenital plane should have enough details so the point selection is done correctly by the user and the homography is correctly computed.

Second Approach

For this reason, and driven by bad results in terms of projections, another cenital plane has been computed starting from zero. In this new approach the scene has correctly been measured by hand and the plane has been done with real measures and high floor detail.

For correctly drawing the plane AutoCAD 2017 software has been used. The second plane approach with all the manual measures extracted from the real scene can be seen in Figure A.2. Figure A.3 represents the final cenital map with the correct camera positions.

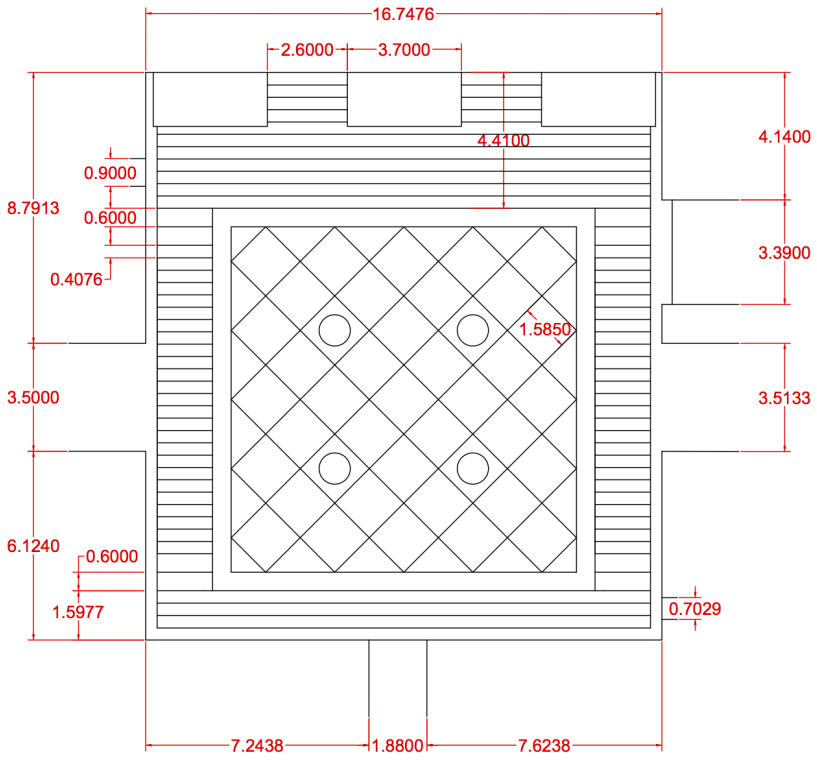


Figure A.2: Second cenital plane approach with real measures

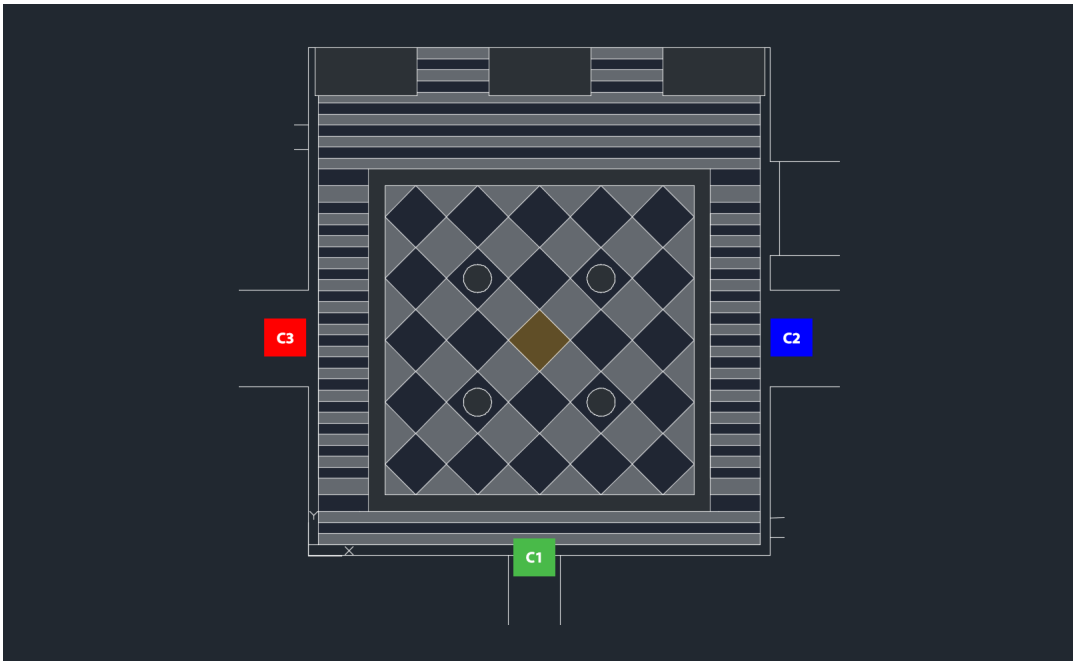


Figure A.3: Second cenital plane approach with camera positions

It is easily observable that differences between Figure A.1 and A.3 are outstanding both in floor details and in general construction proportions. This detail rise leads to a much more easy homography selection points by the user. This is due to the high amount of point options to choose from the new cenital map. Evidentially this means that the final homography matrix, and so, all the projections computed by it, have better accuracy.

Appendix B

AKAZE Point Descriptor

AKAZE detector and descriptor [57] is a fast multi scale feature detection an description approach. It exploits the benefits of nonlinear scale spaces.

Previous approaches such as KAZE [58] or BFSIFT [59] have a main time consumption drawback in terms of nonlinear scale space creation.

Nevertheless, AKAZE uses recent numerical schemes called Fast Explicit Diffusion (FED) [60, 61] in order to build any kind of discretization scheme in a much more faster speed. These FED schemes are embedded in a pyramidal framework in order to achieve the speedup in terms of features detector.

In addition the use of the Modified-Local Difference Binary (M-LDB) descriptor which is described as highly efficient. It exploits gradient and intensity information from the nonlinear scale space. In addition, M-LDB is both scale and rotation invariant.

Appendix C

Parametric Homographies Between Inertial Planes

Homography matrices, as explained during the Thesis, aim to relate the floor plane present in a frame with the cenital view. However, everything that is not exactly in the same plane as the floor is not projected properly when the homography matrix is used. In our work, different semantic are needed to be projected, for instance a door. When using floor homography only its base is correctly projected, whereas the rest of it is disfigured.

In [62] a solution to this problem is proposed. The general idea is to create a multilayer reconstruction. Once the homography matrix ${}^{\pi_{ref-C}}H_{view}$ that relates the image view with the reference frame π_{ref-C} is calculated, one can obtain another matrix ${}^{\pi'}H_{view}$ that relates the same image frame with a parallel plane called inertial plane at a fixed height Δt.

${}^{\pi'}H_{view}$ can be expressed as a function of ${}^{\pi_{ref-C}}H_{view}$ and Δh as described in Eq C.1.

$$ {}^{\pi'}H_v^{-1}(\Delta h) = {}^{\pi}H_v^{-1} + \Delta h P \hat{\mathbf{k}}^T, \tag{C.1} $$

where $P = [u_0\, v_0\, 1]^T$ is the principal point of camera C and $\hat{\mathbf{k}}$ is the unit vector of the Z axis.

All this process is described in Figure C.1.

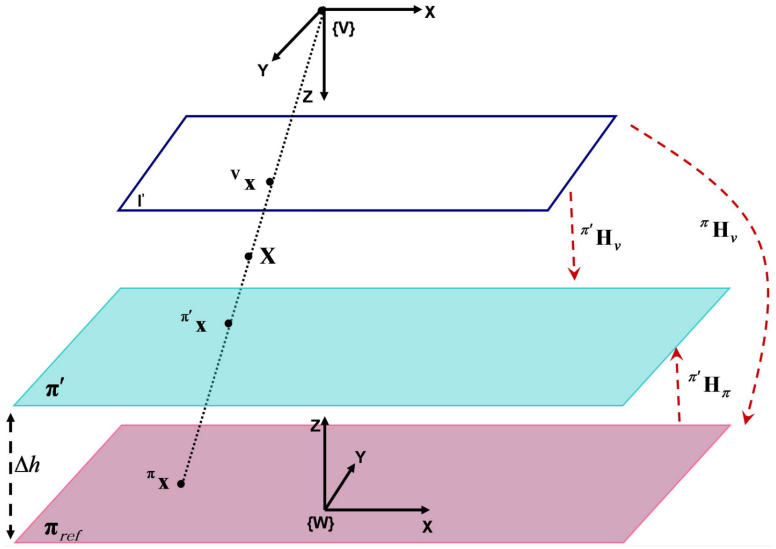


Figure C.1: Extending homography for planes parallel to π_{ref}. ${}^{\pi_{ref}}H_V$ is the available homography between camera view and reference plane π_{ref} .

By this process, ideally a number k of planes could be generate (Figure C.2) in which different object sections are correctly projected. It could lead to a complete semantic map in which all the pixels represent semantic areas that have been correctly projected.

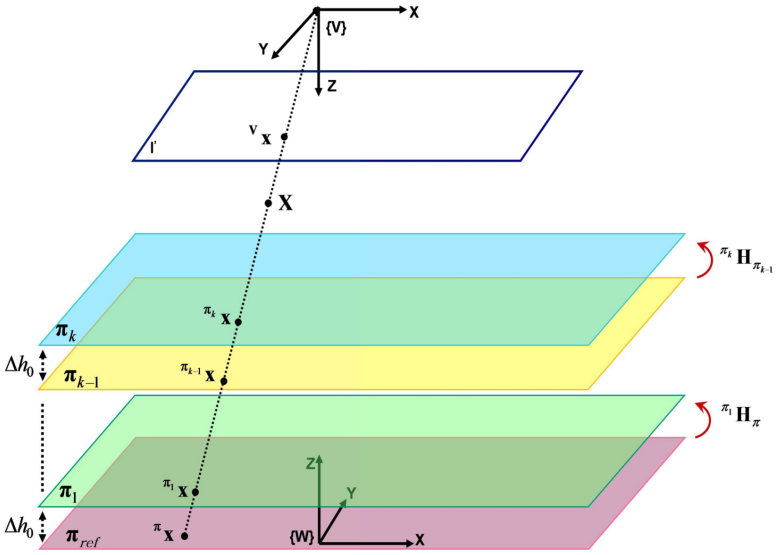


Figure C.2: Set of k inertial planes π_k . Each inertial plane is separated from the other by the same Δh height

Bibliography

[1] A. K. Jain, L. Hong, and Y. Kulkarni, "A multimodal biometric system using fingerprint, face and speech," in *2nd Int'l Conf. AVBPA*, vol. 10, 1999.

[2] X. Wang, "Intelligent multi-camera video surveillance: A review," *Pattern recognition letters*, vol. 34, no. 1, pp. 3–19, 2013.

[3] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[6] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What makes for effective detection proposals?," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 4, pp. 814–830, 2016.

[7] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *CoRR*, 2016.

[8] R. Mazzon and A. Cavallaro, "Multi-camera tracking using a multi-goal social force model," *Neurocomputing*, vol. 100, pp. 41–50, 2013.

[9] A. Turner and A. Penn, "Encoding natural movement as an agent-based system: an investigation into human pedestrian behaviour in the built environment," *Environment and planning B: Planning and Design*, vol. 29, no. 4, pp. 473–490, 2002.

[10] P. Scovanner and M. F. Tappen, "Learning pedestrian dynamics from the real world," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 381–388, IEEE, 2009.

[11] F. Jiang, J. Yuan, S. A. Tsaftaris, and A. K. Katsaggelos, "Anomalous video event detection using spatiotemporal context," *Computer Vision and Image Understanding*, vol. 115, no. 3, pp. 323–333, 2011.

[12] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 304–311, IEEE, 2009.

[13] A. Ess, B. Leibe, and L. Van Gool, "Depth and appearance for mobile scene analysis," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, IEEE, 2007.

[14] C. Wojek, S. Walk, and B. Schiele, "Multi-cue onboard pedestrian detection," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 794–801, IEEE, 2009.

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886–893, IEEE, 2005.

[16] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[17] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, 2014.

[18] Á. García-Martín and J. M. Martínez, "People detection in surveillance: classification and evaluation," *IET Computer Vision*, vol. 9, no. 5, pp. 779–788, 2015.

[19] R. Cutler and L. S. Davis, "Robust real-time periodic motion detection, analysis, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 781–796, 2000.

[20] J. Giebel, D. Gavrila, and C. Schnörr, "A bayesian framework for multi-cue 3d object tracking," *Computer Vision-ECCV 2004*, pp. 241–252, 2004.

[21] K. Okuma, A. Taleghani, N. d. Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," *Computer Vision-ECCV 2004*, pp. 28–39, 2004.

[22] A. García-Martín, B. Alcedo, and J. M. Martínez, "Pdbm: people detection benchmark repository," *Electronics Letters*, vol. 51, no. 7, pp. 559–560, 2015.

[23] K. E. Van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders, "Segmentation as selective search for object recognition," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 1879–1886, IEEE, 2011.

[24] S. Manen, M. Guillaumin, and L. Van Gool, "Prime object proposals with randomized prim's algorithm," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2536–2543, 2013.

[25] P. Rantalankila, J. Kannala, and E. Rahtu, "Generating object segmentation proposals using global and local search," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2417–2424, 2014.

[26] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai, "Fusing generic objectness and visual saliency for salient object detection," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 914–921, IEEE, 2011.

[27] J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 3241–3248, IEEE, 2010.

[28] I. Endres and D. Hoiem, "Category independent object proposals," *Computer Vision–ECCV 2010*, pp. 575–588, 2010.

[29] A. Humayun, F. Li, and J. M. Rehg, "Rigor: Reusing inference in graph cuts for generating object regions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 336–343, 2014.

[30] P. Krähenbühl and V. Koltun, "Geodesic object proposals," in *European Conference on Computer Vision*, pp. 725–739, Springer, 2014.

[31] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 328–335, 2014.

[32] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 73–80, IEEE, 2010.

[33] E. Rahtu, J. Kannala, and M. Blaschko, "Learning a category independent object detection cascade," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 1052–1059, IEEE, 2011.

[34] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "Bing: Binarized normed gradients for objectness estimation at 300fps," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3286–3293, 2014.

[35] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European Conference on Computer Vision*, pp. 391–405, Springer, 2014.

[36] J. Feng, Y. Wei, L. Tao, C. Zhang, and J. Sun, "Salient object detection by composition," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 1028–1035, IEEE, 2011.

[37] Z. Zhang, J. Warrell, and P. H. Torr, "Proposal generation for object detection using cascaded ranking svms," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1497–1504, IEEE, 2011.

[38] M. Van den Bergh, G. Roig, X. Boix, S. Manen, and L. Van Gool, "Online video seeds for temporal window objectness," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 377–384, 2013.

[39] J. Kim and K. Grauman, "Shape sharing for object segmentation," *Computer Vision–ECCV 2012*, pp. 444–458, 2012.

[40] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2147–2154, 2014.

[41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[42] Z. Cai, M. J. Saberian, and N. Vasconcelos, "Learning complexity-aware cascades for deep pedestrian detection," *CoRR*, vol. abs/1507.05348, 2015.

[43] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, 2015.

[44] D. Navon, "Forest before trees: The precedence of global features in visual perception," *Cognitive psychology*, vol. 9, no. 3, pp. 353–383, 1977.

[45] R. A. Rensink, J. K. O'Regan, and J. J. Clark, "To see or not to see: The need for attention to perceive changes in scenes," *Psychological science*, vol. 8, no. 5, pp. 368–373, 1997.

[46] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson, "Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search.," *Psychological review*, vol. 113, no. 4, p. 766, 2006.

[47] A. Torralba and P. Sinha, "Detecting faces in impoverished images," tech. rep., Massachusetts Inst of Tech Cambridge Artificial Intelligence Lab, 2001.

[48] P. Salembier and T. Sikora, *Introduction to MPEG-7: Multimedia Content Description Interface*. New York, NY, USA: John Wiley & Sons, Inc., 2002.

[49] S. Oh, A. Hoogs, M. Turek, and R. Collins, "Content-based retrieval of functional objects in video using scene context," in *European Conference on Computer Vision*, pp. 549–562, Springer, 2010.

[50] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *CoRR*, vol. 1608, 2016.

[51] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, 2016.

[52] Z. Wu, C. Shen, and A. v. d. Hengel, "Wider or deeper: Revisiting the resnet model for visual recognition," *CoRR*, 2016.

[53] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. W. Cottrell, "Understanding convolution for semantic segmentation," *CoRR*, vol. abs/1702.08502, 2017.

[54] A. Miguélez and R. Nieto, "Detección de personas en entornos multicámara utilizando informacion contextural," Master's thesis, Universidad Autónoma de Madrid. Escuela Politécnica Superior, 2016.

[55] J. Xiao and L. Quan, "Multiple view semantic segmentation for street view images," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 686–693, IEEE, 2009.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[57] P. F. Alcantarilla and T. Solutions, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *IEEE Trans. Patt. Anal. Mach. Intell*, vol. 34, no. 7, pp. 1281–1298, 2011.

[58] P. Alcantarilla, A. Bartoli, and A. Davison, "Kaze features," *Computer Vision–ECCV 2012*, pp. 214–227, 2012.

[59] S. Wang, H. You, and K. Fu, "Bfsift: A novel method to find feature matches for sar image registration," *IEEE Geoscience and remote sensing letters*, vol. 9, no. 4, pp. 649–653, 2012.

[60] J. Weickert, S. Grewenig, C. Schroers, and A. Bruhn, "Cyclic schemes for pde-based image analysis," *International Journal of Computer Vision*, vol. 118, no. 3, pp. 275–299, 2016.

[61] S. Grewenig, J. Weickert, and A. Bruhn, "From box filtering to fast explicit diffusion," in *Joint Pattern Recognition Symposium*, pp. 533–542, Springer, 2010.

[62] H. Aliakbarpour, V. S. Prasath, K. Palaniappan, G. Seetharaman, and J. Dias, "Heterogeneous multi-view information fusion: Review of 3-d reconstruction methods and a new registration with uncertainty modeling," *IEEE Access*, vol. 4, pp. 8264–8285, 2016.